DVD ROM

Ubuntu 14.04 – Trusty Tahr

32-bit & 64-bit versions both on one double-sided DVD!

Ubuntu 14.04
BRAND NEW UBUNTU DISTRO

UBUNTU user

EXPLORING THE WORLD OF UBUNTU

COMMAND-LINE TRICKS AND TIPS

SHELL SHOCK!

- Add dialog boxes to your shell scripts
- How to compile Bash scripts
- Good scripting styles
- PowerShell vs. Bash

MINECRAFT
Build (blocky) worlds of adventure

PLUS

Get Things Done with the Nitro task organizer

Monitor your network with LAN Monitoring tools

Discover the power of the Inkscape design software

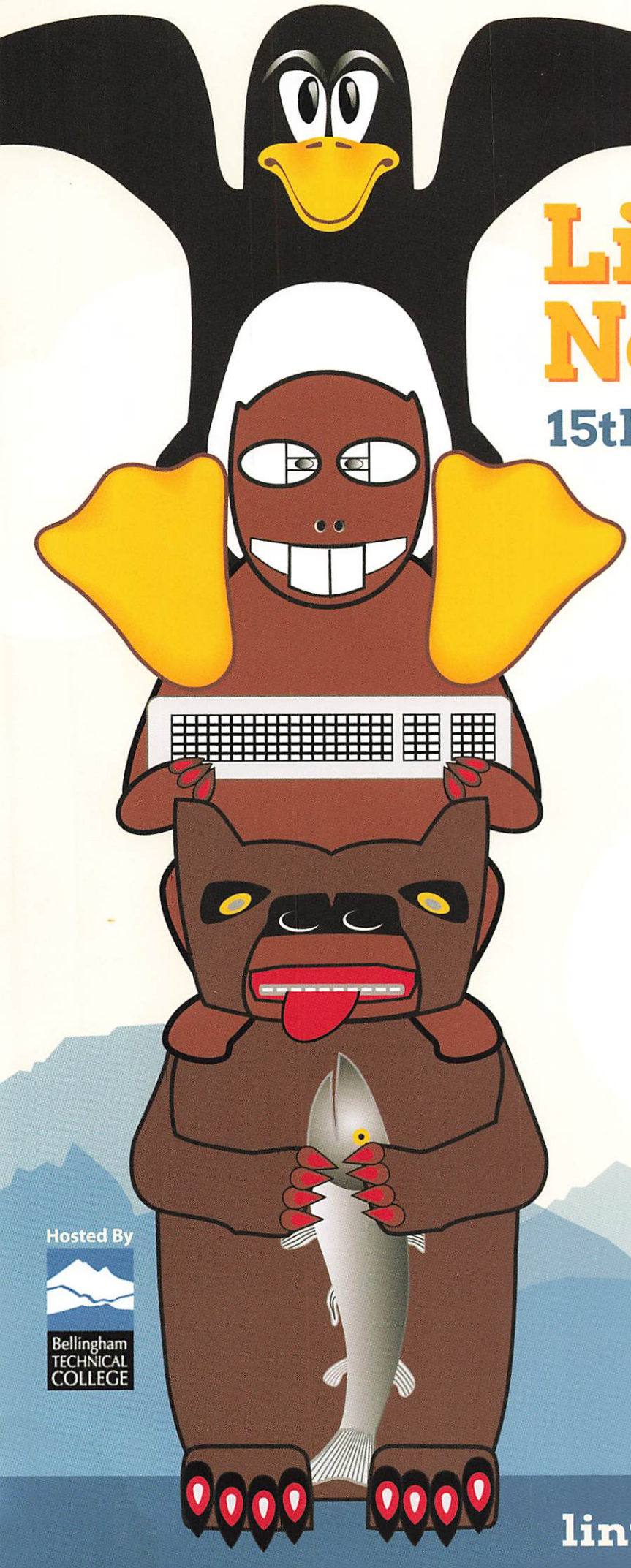
Convert and organize e-books with Calibre

DISCOVERY GUIDE

New to Ubuntu? Check out our special section for first-time users! p. 79

- How to install Ubuntu 14.04
- Ubuntu tips and tricks
- Go online with NetworkManager
- Package management

US$ 15.99
CAN$ 17.99
Summer 2014



LinuxFest Northwest

15th Anniversary

Bellingham, WA
April 26th & 27th

- Grassroots Linux gathering
- Exhibits of all flavors
- Presentations of all levels
- Prizes and after party
- FREE admission & parking
- FREE open source software
- Bring the whole family!

Hosted By

Bellingham TECHNICAL COLLEGE

linuxfestnorthwest.org

My Blue Status Quo

FREE DVD
All new Ubuntu Trusty Tahr on our cover DVD.
See page 7.

Dear Ubuntu User Reader,

If you're new to *Ubuntu User*, welcome (if not, welcome back, y'all). And, if you're new to Ubuntu and the Linux scene in general, you are doubly welcome. Maybe you're here because of all the XP hoo-ha and have decided to give this Open Source/Free Software stuff a go and see what the fuss is about.

First, a warning: You're going to be confused. Linux does things differently from Windows. Not radically different, mind. An operating system is an operating system, and they all have files, directories, and applications that do more or less the same things. But, Linux is different enough to be unsettling and make you feel that you're productivity is being undermined. You might think, why can't Linux be more like Windows, because the Windows way is the tried, tested, and trusted way of doing things, right? Well, no. *NIX type operating systems existed much before Windows, so it is Microsoft that decided to diverge from the norm. But, before you give up, consider this: People like things they're used to. So do you. It's not that Linux makes things difficult, it is that you're not used to how Linux does things. Stick with it, you'll figure it out. And, you'll like it when you do.

Second, kind of related to the first: You're going to wonder why you can't run your Windows stuff on Linux (okay, maybe you can, but it is not all that straightforward, and I would advise against it). The answer is: You're not meant to. Linux has plenty of perfectly adequate applications that can stand in for whatever you used under Windows and then some.

Despite the above, you may think that not being able to run Microsoft products is a hurdle, a major stumbling block in the way of a larger Linux adoption. If Linux were "Windows-compliant," so to speak, it would make usage easier for all those Windows refugees, and there can't be anything wrong with that, can there? But, what if I told you getting more users to use Linux is not ultimately the aim of the game? At least not at any price. But, you may argue, more users will put more pressure on manufacturers to build more and better software for Linux, like drivers for video cards and printers and such. Which would make life easier for users, which in turn would attract more adopters, right?

Unfortunately, as Android (a massively used Linux) has taught us, having more users does not subvert the status quo, and the status quo is in desperate need of being subverted.

This needs some explaining.

I'm not going to disparage Microsoft today. I have disparaged them elsewhere far too much. Let's talk about Adobe instead. Adobe is best known for being the developers of Photoshop, one of the programs most required by new Linux users. If that's all you know about Adobe, allow me to fill you in. You may be aware that Adobe produces Flash. Flash (old-timers will remember this) was originally invented by a company called Macromedia. Flash was acquired by Adobe when they absorbed Macromedia in 2005. Before that, Adobe had weakened Macromedia with several spurious lawsuits. One of them was because Macromedia had dared to use a tabbed interface in Fireworks, an application for creating web-oriented graphics. Adobe's aim was to push Macromedia to bankruptcy because of FreeHand, a competitor of Adobe's Illustrator. When rumors of a takeover on behalf of Microsoft started circulating, Adobe didn't waste any time swooping in and buying Macromedia and immediately killing Freehand.

They also more or less killed Quark (although Quark is still around, it is a shadow of its former self) by bundling their own layout application, InDesign, with the rest of their software, creating to all practical effects a designer's operating system within an operating system. For the record, Quark Xpress was, up until Adobe used dumping tactics to corner the market, the gold standard for layout. Never heard of it? See how well Adobe's strategy worked?

Whether or not you agree with my appreciation that these sort of tactics are bad for consumers, the empirical truth is that Adobe has used them to exterminate all competition – something that opens users to all sorts of abuse because now there's only one provider you can go to for enterprise-grade design applications. Or, so they would have you believe. These types of practices are commonplace within the proprietary software industry and, therefore, going from Windows to Linux and ending up using the same closed software achieves nothing – in no way are users empowered, nor are dirty, uncompetitive policies that lead to de facto monopolies curtailed.

You may counter with the adage that competition is a good thing, that maybe Free Software app developers should try competing and quit whining. Yes, competition is great, but only on a level playing field. Small teams of voluntary developers can't leverage patent lawyers or lobby governments.

Adobe has proven that they can and will. What would happen if Adobe set its sights on Linux? They'd use their market share might to stomp all over free design programs, that's what. Linux has GIMP, Krita, Inkscape, Blender, Scribus, and a few other excellent design programs. These applications more than make up for the lack of support of Adobe products on Linux. If Adobe decided they were a threat, without user support, they'd be doomed.

So, again, if you're new, enjoy your stay, but before you air your views on what you consider is best for Linux, give Free Software applications a chance, and then we'll talk.

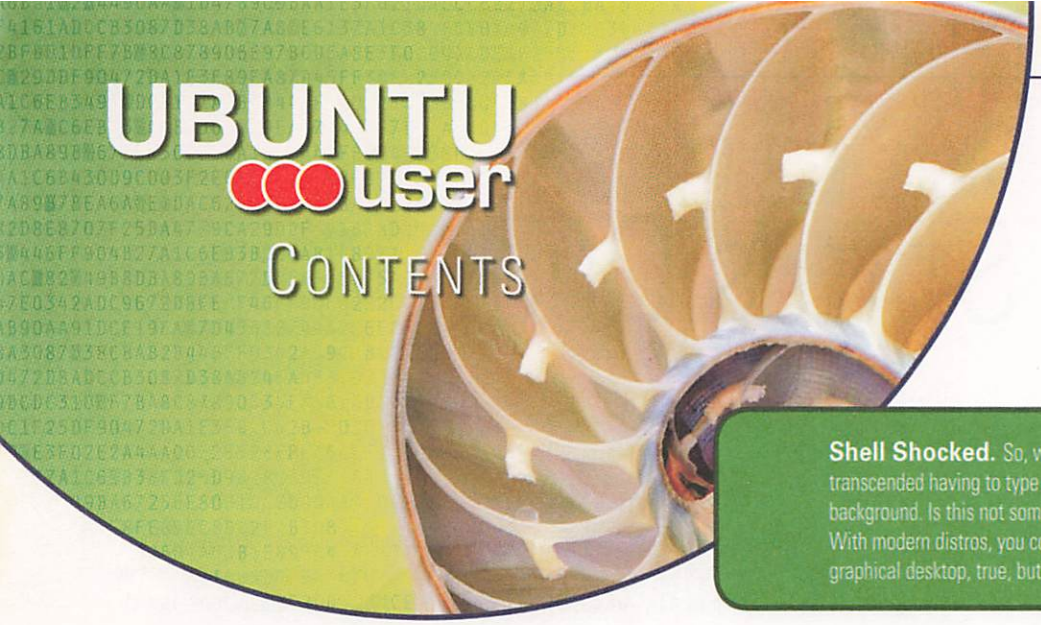
Disagree? Do you think Linux should be able to run more proprietary Software? Send me your views and reasoning on the matter to pbrown@ubuntu-user.es.

Paul C. Brown,

Editor in Chief

UBUNTU user

Contents



Shell Shocked. So, what's the deal? Surely we have transcended having to type obscure commands on a black background. Is this not something reminiscent of the bad old days? With modern distros, you could never leave the safe haven of your graphical desktop, true, but you'd definitely be missing out.

Features

09 The Linux Shell Is Your Friend
The shell may look like an old-fashioned bit of technology, only useful for the Linux hardcore programmers and system administrators, but knowing a few commands and how to link them together goes long way.

10 PowerShell vs. Bash
Bash and PowerShell are related, but as in all large families, the branches of the tree often take on different forms.

14 Writing Clearer Bash Scripts
Does the ingenious script your wrote last month look like confusing spaghetti code today? We show some proven style guidelines that can provide order and readability.

20 Shell Script Dialogs
From simple queries to complex menus: Using dialog, you can create a graphical interface for shell scripts with only a few extra lines of code.

28 Compiling Scripts
The Shell Script Compiler converts scripts into binaries, which protects against accidental changes but also carries some pitfalls.

Plus...

03 Editorial
If you're an Ubuntu noobie, read this.

06 Ubuntu 2014
Jono walks us through what's new in Ubuntu 14.04 - Trusty Tahr.

97 Contact Info / Authors

98 Next Issue

On the DVD

08 Ubuntu 14.04 - Trusty Tahr, 32 & 64 bits
The two most useful Ubuntu versions on one DVD

- Unity support for High-DPI screens
- Linux kernel 3.13
- Improved thermal management
- New power management system
- LibreOffice 4.2
- Smarter searches
- Seamless social media integration

Know How

42 **Bandwidth Monitoring**
Many programs communicate over the network, and when a bottleneck occurs, these tools can help you determine the cause.

31 **Off-Road Navigation**
Anyone doing off-road navigation needs special maps. You can create these maps in no time at all with some specialized tools.

48 **Systemd**
Some Linux followers see systemd as the best thing since sliced bread – for others, it's the work of the devil. However, it also has the stuff to clear out old trenches and form a unified base for Linux.

54 **E-Books with Calibre**
Calibre lets you bring some order to your digital library. You not only can restore reading material lost on your hard drive, but you can also convert digital books into a format of your choice.

Desktop

58 **Banking Applications**
"Where does all my money go?" A good personal finance management app like Buddi or HomeBank can help you answer this question.

62 **Organize Tasks with Nitro**
Nitro is an application with a modern design created for managing task lists. This free program synchronizes with different platforms on all your devices, even your smartphone.

66 **Minecraft**
With the current version 1.7, Minecraft developers have dared to take a step away from the popular blocky look. However, the real thrill of the game is hardly its visual effects.

71 **The QupZilla Web Browser**
QupZilla is a new Linux web browser that enters an already crowded arena. But, QupZilla holds its own when it comes to speed and ergonomics. We put the browser through its paces in our lab.

74 **Creative Inkscape Effects**
The Inkscape vector graphics program not only lets you work with pencil and ink-like lines, it can also help you mimic the elegant brushstrokes of an oil painting or simulate water colors. Our three easy-to-follow examples show you how to paint your masterpiece.

Community

92 **Q&A with Stuart Langridge and Bryan Lunduke**
Stuart Langridge and Bryan Lunduke, team up (more or less) to answer all your Ubuntu questions. This month: My Very Own OS and Office Space.

94 **Developing Free Hardware**
We look at Aaron Seigo's projects Vivaldi and Improv as examples and discover how hard it really is to manage a truly free hardware project.

Discovery Guide

If you are new to Ubuntu, these timely tutorials will help you get started.

80 Installing Ubuntu 14.04
Ubuntu 14.04 offers a few changes to the installation process. We'll show how to upgrade or install the latest release.

84 Network and Updates
NetworkManager lets you access the Internet during the Ubuntu installation, but first download updates to make your system more secure.

86 Package Management
Ubuntu's Software Center lets you install just about anything, including free software, purchased apps, and games.

89 Multimedia
You can configure your Ubuntu system to support movies, music, and Internet telephony. We present some favorite multimedia apps..

Jono shares what's inside Ubuntu's latest Long-Term Support release.

Ubuntu 14.04

Latest LTS

BY JONO BACON

As I write this, it is the Monday evening before we release Ubuntu 14.04 on Thursday.

As many of you know, Ubuntu 14.04 is our next Long-Term Support (LTS) release. Every two years, we nominate an Ubuntu release as an LTS that is provided with an extended period of support. The rationale for this approach is that many people delivering Ubuntu in the enterprise don't want to update their entire deployment every six months. Updating once every two years is more comfortable, so people use the LTS, which has security updates for a longer period of time.

We tend to take a slightly more conservative approach to these LTS releases and are usually more reluctant to drop large new pieces of technology into the platform. The main reason is not just that we want to assure quality for our users, we also don't want to cause significant headaches for these enterprise deployments when big pieces of the platform change. This is one benefit of releasing every six months – we can drop new pieces of technology into the non-LTS releases, so the IT staff can get to know the changes before the LTS upgrade period comes around.

So, what's in Ubuntu 14.04? Well, that depends on which version of Ubuntu you're using. As we move toward our convergent future, we have different features in the desktop, server, and mobile devices.

Let's start with Desktop. On the desktop side, we are still shipping our proven Unity 7 and X.org code-base along with a raft of version upgrades to the software we ship by default. As part of the desktop, we've also added a new lock screen, window management changes and improvements, high DPI support, refined web apps support, and more. It is rock solid.

On the cloud and server side, much of our effort has gone into cloud orchestration tools. This includes Juju, which can deploy to a range of different clouds in a heartbeat and pull from the diverse charm store, which is packed with individual components like Django and WordPress, as well as bundles of charms for making deployment even easier and more flexible.

As we march toward the highly anticipated Ubuntu for smartphones and tablets, we have made significant improvements. Ubuntu 14.04 will see the first version of the tablet as well as the next iteration of the phone.

Because of the nature of Ubuntu's convergence model, everything going into the new code-base benefits both phone and tablet users. This includes completely redesigned and refined scopes, push notifications, a raft of new applications, Google contact/calendar syncing, new multi-tasking features, and much more. We have also significantly refined our SDK and developer platform with HTML5 improvements, an Ubuntu emulator, a completely redeveloped developer.ubuntu.com and more.

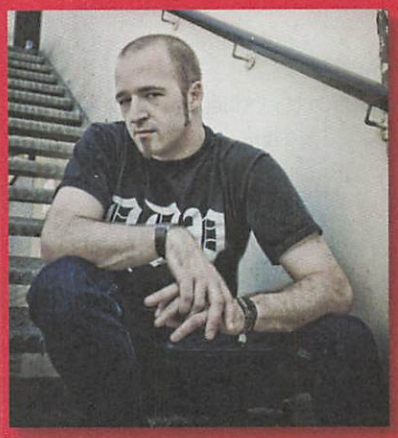
For those of you excited about the future convergent desktop coming in the 14.10-15.04 timeframe, Ubuntu 14.04 on the desktop will also ship a Unity 8 "preview" session. This lets you select Unity 8 from the login screen and run the Unity 8 tablet mode on your desktop, all running on top of the Mir display server (which is already powering the phone and tablet platforms).

To be very clear, this is a *preview* session, and it won't look like a desktop; we will indeed be making Unity 8 look like a desktop as we move toward 14.10 and 15.04. But, don't worry, we don't plan on making our desktop look like a tablet. Ahem, Microsoft.

So, there you have it, many exciting things are in store across desktop, cloud, phone, and tablet. On the desktop and server side, we will provide everything as a rock-solid LTS release; and, on the mobile and tablet side, we'll continue to build out the platform.

Fortunately, you will be able to buy the very first Ubuntu phones from either BQ or Meizu later this year. If you want to see what all the fuss is about right now, you can get a Nexus 4 phone or a 2013 Nexus 7 tablet and install Ubuntu today. If you have a spare device, give it a whirl!

Before I wrap up, I want to say a huge "thank you" to every single one of you who continues to believe in and support Ubuntu. We have made some bold and innovative decisions to make Ubuntu the world-class platform we believe it can be, and you folks have stood behind us all the way. Here's to a prosperous Ubuntu future! •



Jono Bacon

Jono Bacon is a leading community manager, engineering manager, consultant, and author. Currently, he works as the Ubuntu Community Manager at Canonical, leading a team that grows, inspires, and encourages the global Ubuntu community.

Maker Faire®

YEAR OF THE MAKER

GOLDSMITH SPONSORS

intel

radioshack

SILVERSMITH SPONSORS

AUTODESK.

Cognizant

ITA

ITALIAN TRADE AGENCY

ICE - Italian Trade Commission

Trade Promotion Section of the Italian Embassy

Ubuntu 14.04 – Trusty Tahr, 32 & 64 bits

On the DVD

Ubuntu Software Center
Ubuntu User Issue 12 - Screenshot
Ubuntu 11.10
Packed with Bonus Software
ubuntu 11.10 Oneiric Ocelot
UBUNTU user
EXPLORING THE WORLD OF UBUNTU
CANONICAL'S COOL NEW CLOUD TOOL
RUNNING ON JUJU
12 Great Games
Head-Up Display: Is HUD the future of the menu?
5GB of free online storage waits for you at Ubuntu One
Tethering: Connect your laptop to a smartphone for instant Internet
Close

This month, on our double-sided disk, we have the newest version of Ubuntu, 14.04 – Trusty Tahr. Trusty is the most mature and stable version of Ubuntu to date and comes with the latest and greatest Free Software.

Side A comes with the 32-bit version of Trusty, ideal for older machines or if you have compatibility problems with libraries or apps.

On side B, you'll find the 64-bit version of Ubuntu. This will work better on more modern machines and will deliver a faster and more optimized experience. If in doubt, try this version first.

Ubuntu 14.04

- Unity support for High-DPI screens and desktop scaling
- Linux kernel 3.13
- New and improved thermal and power management
- LibreOffice 4.2
- Smarter searches with more filters and lenses
- Seamless social network integration

Double-Sided DVD

This DVD is a double-sided disc. A DVD device always reads the side of the disc that is facing down. This DVD is labeled conventionally – the label you see as you place the disc into the tray applies to the data on the opposite side. If you are installing Kubuntu, the Kubuntu label should be facing up in the tray.

DEFECTIVE DISC?
Defective discs will be replaced. Please send an email to cs@ubuntu-user.com.

Additional Resources

[1] Ubuntu website: *http://www.ubuntu.com*

The shell may look like an old-fashioned bit of technology, only useful for the Linux hardcore programmers and system administrators, but knowing a few commands and how to link them together goes long way.

The Linux shell is your friend

SHELL SHOCKED

The Linux shell is peculiar old thing. It was developed in the '70s for UNIX computers, and its look and feel has not change much in its forty years of existence. So, what's the deal? Surely we have transcended having to type in obscure commands on a black background. Is this not something reminiscent of the bad old days when floppy disks could hold only a few hundred bytes of data and phosphor green monitors burnt your eyeballs right out of their sockets?

In modern Linux distros, such as Ubuntu, you can spend your whole life within your graphical desktop and not have to deal with a command line once. You can download software, adjust configurations, copy and move files, and start and stop system processes by clicking, dragging, and dropping. What's the point, then, of such an arcane tool?

You could never leave the safe haven of your graphical desktop, true, but you'd definitely be missing out. When we worked on *Linux Magazine* Spain, from time to time – every year more or less – we produced a special Shell edition that brought together the best articles on shell usage, as well as new material we hadn't been able to fit into other issues. The shell issues were invariably a hit. They would outsell regular issues three to one, and people would order them from us well after they had been taken off the newsstands.

Who bought them? Surely not the old guard. Despite all our efforts, it was very difficult to come up with tricks a weathered *NIX sys admin was not already aware of. No, these were end users. We knew because they emailed us questions and suggestions for future issues; they phoned us to clear up some obscure point. We even got letters, yes, handwritten letters, in envelopes, from several prisons requesting copies for inmates trying to further educate themselves. All these people did not choose to learn about the shell because the desktops were inadequate. The developers of Gnome, KDE, and Unity have done their utmost, bless their penguin-printed knitted socks, to make the shell obsolete.

But no matter how beautiful and feature-packed Linux desktops become, many users – a great many users – still want to learn about and use the shell. There may be a romantic component to it. It looks more like hacking if you're typing green letters on a black background, even if you're only copying files onto a thumbdrive. If we could only get the BZZZZZRT! they dub over in movies and TV shows every time a line of output appears, shell usage would go mainstream in a hurry.

Yet, there's more to it than that. The shell has several clear things going for it. For one, it's universal: Every Linux has it and most tools are the same, regardless of distribution. It's also accessible from anywhere. You can SSH from your laptop into a shell on your local network and start mending, updating, restarting, and configuring right away. The same goes with a machine on the other side of the world, or, instead of from your laptop, from your smartphone. SSH and a shell interface are engineers' gift to sys admins who would prefer not to get out of bed at 4am to restart a service that has crashed at work.

Despite rewrites and improvements, shells remain dependable, compatible, and consistent. Your graphical environment can freeze and the keyboard lock up, but chances are that you can still wiggle your way in via a shell and stop whatever process is messing with your workday. Plus, you can pick up a manual about *NIX administration, and it won't really matter if it was published in the '80s or last month, most of the commands and scripts will still work.

Modern shells also pack a punch. This cannot be stated enough. From where else could you back up and change the format of all the music and sounds in your hard disk, download a whole website including graphics, or harvest every single email from every single text document with one single (and short) line of code? Don't get me started on when you string commands together in scripts.

So, this month we're giving it up for the shell – a powerful thing you'll learn to love.|

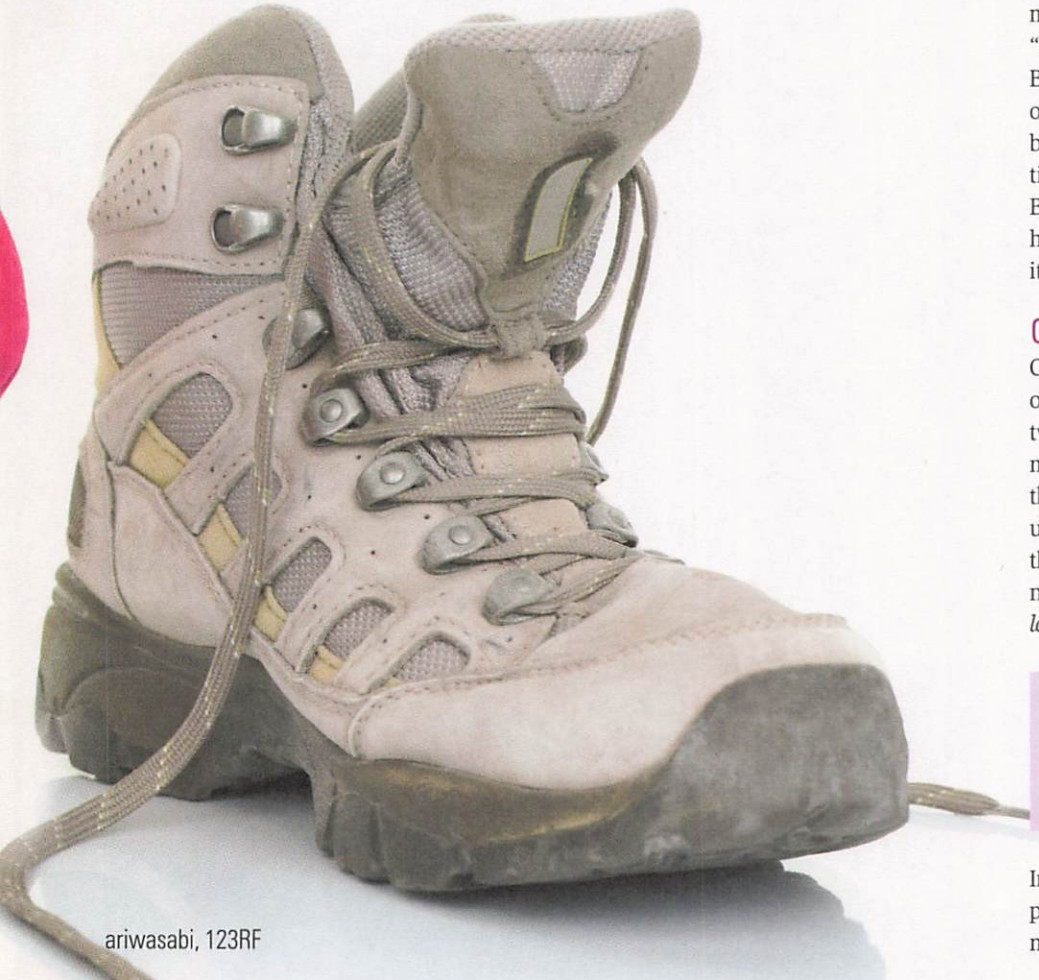
Bash and PowerShell are related, but as in all large families, the branches of the tree often take on different forms.

Comparing Windows PowerShell and Bash

Disparate Development

BY MARCUS NASAREK

Several years have passed since my previous article looking at the command-line languages under Windows and Linux [1] – time to reinvestigate how these two opponents have developed over the past half decade. A comparison of Windows PowerShell and Bash might indicate who has the current edge in modern scripting.



ariwasabi, 123RF

Windows PowerShell was a completely fresh start of Microsoft's command language under Vista (see the "New Beginning" box). Meanwhile, Bash under Linux was always at the core of the system (see the "Tried and Tested" box) with the project participants continuously updating the interface. So, has Bash matured and got better with age, or has inherited too much cruft that weighs it down?

Comparing Concepts

One of the most important characteristics of a scripting language is the ability to link two commands, assuming that one command receives, understands, and processes the output of the other. As a rule, Bash uses plain text as input and output. Thus, the following command shows the size and name of the three largest files in the */var/log* directory:

```
$ ls -l /var/log | sed 's/ \+/,/g' ➥
  | cut -d',' -f 5,9 | sort -g | tail -3
267453,core.log
443742,core.log.1
584584,lastlog
```

In Windows PowerShell, the input and output are objects; thus, entire data structures migrate from command to command,

which assumes that objects follow certain guidelines. To get the same result in Windows as in the previous example, you need to know the properties of the corresponding objects:

```
PS> Get-ChildItem -Directory ➥
  | Sort-Object -Property Length ➥
  | Select-Object Length,Name -Last 3
   Length Name
   ------ ----
 597379 Manual.pdf
 5715026 Applications.in.Ruby.pdf
 27318109 AnnualCatalogue_de-EU.pdf
```

The fact that Bash presents output as text naturally makes it easier to pass information on to the next program. Most things can be worked out by trial and error – which is nearly impossible in PowerShell.

In the example, you need the properties Length and Name. Intuitively, you might have used Size instead of Length. So, to know the properties of an object, sometimes the documentation is the only resort.

As it is, the example from PowerShell 3.0 is already easier than in previous versions. In earlier versions, the *Get-ChildItem* cmdlet (which corresponds roughly to ls) didn't yet recognize the *-Directory* option. Instead, it depended on a subsequent cmdlet (*Where-Object {-not $_.PSIsContainer}*) to be built into the pipe to filter by directory, because only directories have the property PSIsContainer.

The difference in handling input and output is just one of many differences between Bash and PowerShell, but it's strikingly obvious. The difference in concept is basically that Bash aims at division of labor while PowerShell focuses on tasks, which was reinforced further in versions 3.0 and 4.0.

Steep Learning Curve

Bash uses only about 40 internal functions and 100 helper programs. PowerShell 4.0 has 299 built-in cmdlets. Version 2.0 (Windows 7) already had 251. Every cmdlet has options, which increase in number with each new PowerShell version.

Thus, the Get-Content cmdlet in Version 3.0 (Windows 8) has the option *-Tail <count>*, which the developers apparently borrowed from the Unix tail command. Microsoft provides a quick reference guide to the new PowerShell features [4].

The cmdlets are more like mini-programs compared to the Bash built-ins, which becomes obvious when the cmdlet does not do what you expect it to do. When this happens, which is often, you'll need to look at the reference document to find the correct parameters and options. On the plus side, cmdlets can execute more complex tasks in one call. For example, to call up all the links on the *ubuntu-user.com* website, all you need in PowerShell 3.0 are the commands in Listing 1.

Sometimes this task orientation seems a bit much, and it makes certain tasks a bit cumbersome. To generate a password in

Listing 1: Call Up Links with PowerShell

```
PS> (Invoke-WebRequest http://www.ubuntu-user.com/).links


innerHTML : <IMG border=0 alt=logoOL.gif src="http://www.ubuntu-user.com/pix/logoOL.gif"
width=140 height=62>
innerText :
outerHTML : <A href="/"><IMG border=0 alt=logoOL.gif src="http://www.ubuntu-user.com/pix/
logoOL.gif" width=140 height=62></A>
outerText :
tagName   : A
href      : /
```

New Beginning

Microsoft has wanted that the graphical interface to be all you need to operate a computer. There was simply no room for an unadorned shell. Hence, Windows command line sat for decades in a hidden niche.

Despite this decision, the system still relied on scripts for management and, in consequence, many interpreter languages soon sprouted up – Windows Scripting Host (WSH), the Windows Management Interface (WMI), HTML Applications (HTA), the Active Directory Services Interface (ADSI), and Visual Basic Script (VBS).

Microsoft then set the system on a totally new course with Windows Vista. Desktop and server editions began sharing the same code base. Developers also reworked the system architecture and based everything on reusable objects. The .NET framework was supposed to be the unifying structure for all modern applications and provide the building blocks for all new ones. An interpreter should be able to access the system library.

The new shell should, therefore, have all the qualities that modern interpreters such as Python and Ruby already had, except now on Windows as well. Hence PowerShell is object-oriented. It can pass the output from one command in a pipe to another command, and .NET components can be integrated. In terms of modern scripting languages, this was hardly a breakthrough, but it proved to be a big step for Microsoft toward professional system administration.

Four generations of PowerShell have emerged from the company since 2006. Windows 7 includes PowerShell 2.0, and Windows 8 has PowerShell 3.0 with many enhancements. The current version 4.0 was written for Windows 8.1 and, with .NET Framework 4.5.1, also comes as an update for Windows 7 and Windows 8.

Tried and Tested

Bash development is rather straightforward in comparison. As many as 111 enhancements have been spliced into Bash [2] since my last evaluation in 2007 – the largest number in the jump from version 3.2 to 4.0 in 2009. The comparatively minor changes in the current version 4.2 prove how well Bash has matured.

Just under half (18 of 39) of the built-in Bash functions are from the Bourne Shell. The cd and test are certainly among the most commonly used, but they also count among the relics. The other half of the functions, among them echo and read, came along with the Bourne Again Shell. They handle everyday tasks, with Bash leaving the special ones to other programs.

Splitting the workload has always been a well-known Linux concept. The login shell serves more as a central command function for the many specialized programs. Thus, nearly 100 programs are available in the GNU Coreutils, which comes with just about every Linux distribution [3].

Bash, all you need is to read the file descriptor for the random number generator and prepare it somewhat (see Listing 2).

To accomplish the same in PowerShell 3.0, you can load a system library from .NET Framework and use a module from it. In the Assembly System.Web, you'll find the Security.Membership module and its GeneratePassword function (Listing 3). An assembly is similar to the Shared Object principle in Linux. You can load assemblies into scripts and use their functions from there.

PowerShell may leave you wondering things like how to determine which objects use which methods, and in which of the many thousand system libraries can you find the function GeneratePassword. PowerShell definitely requires some heavy lifting. You need to study the extensive reference to the .NET Framework [5] to get anywhere.

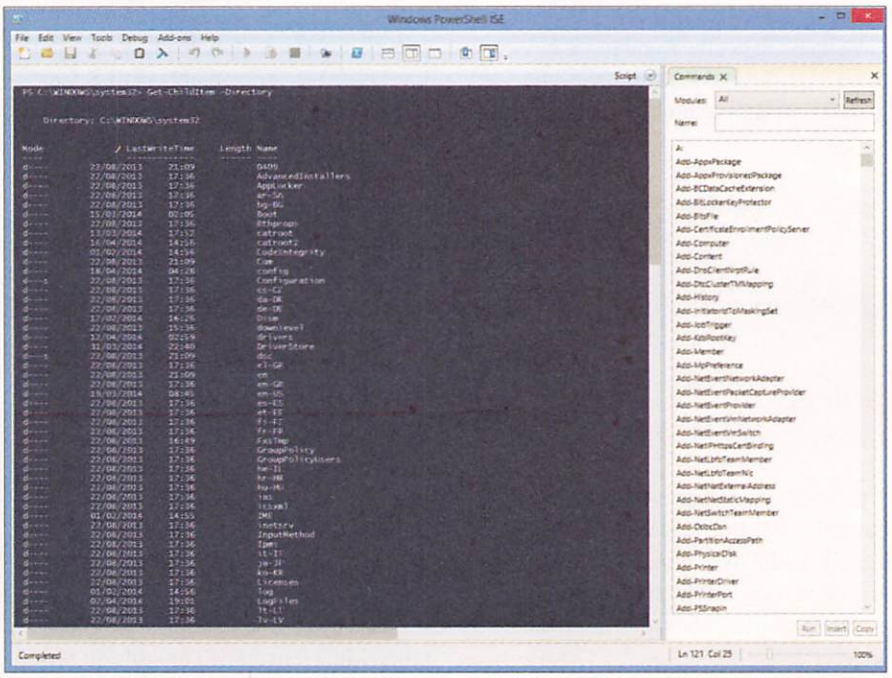


Figure 1: The PowerShell ISE development environment is part of the PowerShell installation.

Assistance

To get command help in Bash, you can just enter *apropos <keyword>*, which lists all commands that include the given keyword. To find out which options a command supports, simply enter the command without options. To get to know Bash even better, a look at the Bash Reference also helps [6].

There is indeed a *Get-Help* cmdlet in PowerShell, but it doesn't help beginners in dealing with slightly more complex objects. Microsoft therefore provides a small development environment for PowerShell that includes an interactive overview of the commands and lets you explore the cmdlet properties to be able to test, activate, or configure them (Figure 1).

The PowerShell Integrated Scripting Environment (ISE) proves to be quite useful in that it not only makes working with cmdlets easier, but it also includes a debugger that executes the scripts step by step to help in troubleshooting. You can also use ISE to start a PowerShell on a remote computer, similar to SSH in Linux. For this to work, the administrator needs to allow remote access emitting the *Enable-PSRemoting -Force* command, which evokes the WinRM service to provide the access.

The question of a development environment doesn't even come up for many Linux admins in that powerful editors such as Vim or Emacs have many functions that facilitate scripting, such as syntax highlighting, complex functions to navigate scripts, and built-in macros. Editors of this kind have not as yet become standard Windows features.

Conclusion

The examples I have given show how different the shell concepts are. Aliases such as ls, mkdir and ps might make the changeover of admins from Unix to Windows easier, but it's doubtful that this will make any difference in the end.

PowerShell shows its strengths mainly in administering Microsoft server applications. The cmdlets are designed to execute tasks with a minimum of commands. The reason entire objects are used to communicate instead of plain text is because of the underlying framework more than to simplify administration.

Bash takes a completely different approach. Shell scripts don't manage products but rather a versatility of tools. For more complex tasks, you can use specialized interpreters such as Perl and Ruby. The main purpose of Bash is to manage systems. •

Listing 2: Generate a Password with Bash

```
$ echo $(< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c10)

Kc-TUGC8cd
```

Listing 3: Generate a Password with PowerShell

```
PS> $Assembly = Add-Type -AssemblyName System.Web

PS> [System.Web.Security.Membership]::GeneratePassword(10,3)

lGv%.ua]Wl
```

Info

[1] "Comparing Bash with the Windows Vista shell" by Marcus Nasarek: *http://www.linux-magazine.com/Issues/2007/78/Bash-vs.-Vista-PowerShell*

[2] News for Bash: *http://tiswww.case.edu/php/chet/bash/NEWS*

[3] GNU Coreutils: *http://www.gnu.org/software/coreutils/manual/coreutils.html*

[4] Windows PowerShell 3.0 and Server Manager Quick Reference Guides: *http://www.microsoft.com/en-us/download/details.aspx?id=30002*

[5] General Reference for the .NET Framework: *http://msdn.microsoft.com/en-us/library/vstudio/sxe8hcf2%28v=vs.100%29.aspx*

[6] Bash 4.2 Reference: *http://www.gnu.org/software/bash/manual/html_node/index.html*

REAL SOLUTIONS FOR REAL NETWORKS

FREE CD or DVD in Every Issue!

Each issue delivers technical solutions to the real-world problems you face every day.

Learn the latest techniques for better:

- network security
- system management
- troubleshooting
- performance tuning
- virtualization
- cloud computing

on Windows, Linux, Solaris, and popular varieties of Unix.

6 issues per year!

ORDER ONLINE AT: shop.linuxnewmedia.com

Does the ingenious script your wrote last month look like confusing spaghetti code today? We show some proven style guidelines that can provide order and readability.

Writing clearer Bash scripts

Right Ingredients

BY TIM SCHÜRMANN

Bash scripts help rename files, sort a large address list, or execute other similarly tedious tasks. Scripts created under pressure can often look garbled (see Listing 1 for a prime example). Quick hacks may have their place, but they are often confusing and might be totally unclear a week later. For this reason, many users don't share their scripts, feeling they are not up to par. All this contributes to making such scripts hard to expand or change down the road.

Regulator

For these reasons it's best to structure, format and, above all, comment your Bash scripts from the outset. Some guidelines can be helpful, such as the one from Google [1] or the handy PDF booklet written by Fritz Mehner [2] that you can print and keep by you computer (Figures 1 and 2). Whether you choose a commercial style guide or one from academia, the basic guidelines are the same.

Your script should always identify the shell it is using at the beginning. If the script uses Bash functions, include *#!/bin/bash* on the first line. That way, systems that don't use Bash as the default shell, but have it installed, can still use it.

Fresh Air

To make the script more readable, get used to using the proper indentation and including blank lines. The confusing elements in Listing 1 become less cryptic if presented as in Listing 2.

Each line has a separate instruction and a blank line separates blocks of related code, such as the block for the *for-do* loop. The statements in a block should be indented, such as the *do* portion of the loop. If you use space characters or a tab for indentation, different editors might interpret them differently. Style guides almost unanimously suggest using a double space character in place of a tab.

Notice that the *do* command is by itself on a line. Google, however, suggests including it on the previous *for* line. The same goes for *while* loops and the *then* in an *if* block. The *else,* however, is often best placed on its separate line, as is the ending

Natalia Klenova, 123RF

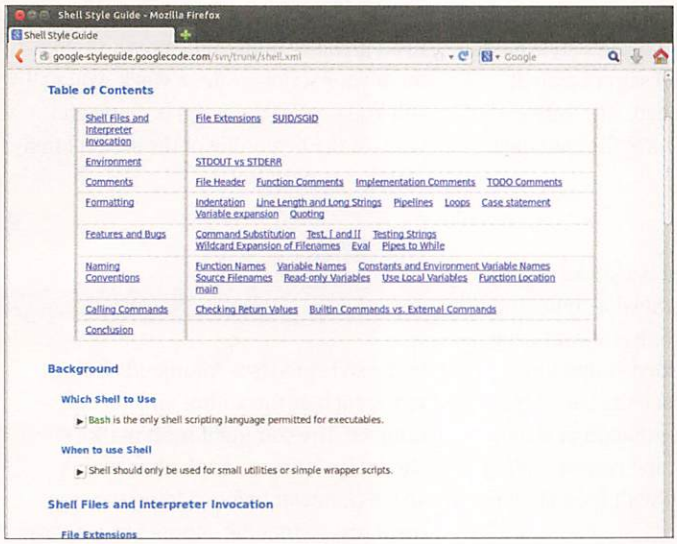

Shell Style Guide - Mozilla Firefox
Shell Style Guide
google-styleguide.googlecode.com/svn/trunk/shell.xml
Table of Contents
Shell Files and Interpreter Invocation
File Extensions SUID/SGID
Environment
STDOUT vs STDERR
Comments
File Header Function Comments Implementation Comments TODO Comments
Formatting
Indentation Line Length and Long Strings Pipelines Loops Case statement Variable expansion Quoting
Features and Bugs
Command Substitution Test, [and [[Testing Strings Wildcard Expansion of Filenames Eval Pipes to While
Naming Conventions
Function Names Variable Names Constants and Environment Variable Names Source Filenames Read-only Variables Use Local Variables Function Location main
Calling Commands
Checking Return Values Builtin Commands vs. External Commands
Conclusion
Background
Which Shell to Use
Bash is the only shell scripting language permitted for executables.
When to use Shell
Shell should only be used for small utilities or simple wrapper scripts.
Shell Files and Interpreter Invocation
File Extensions

Figure 1: You get details on individual guidelines at Google by clicking the triangle in front of a guideline.

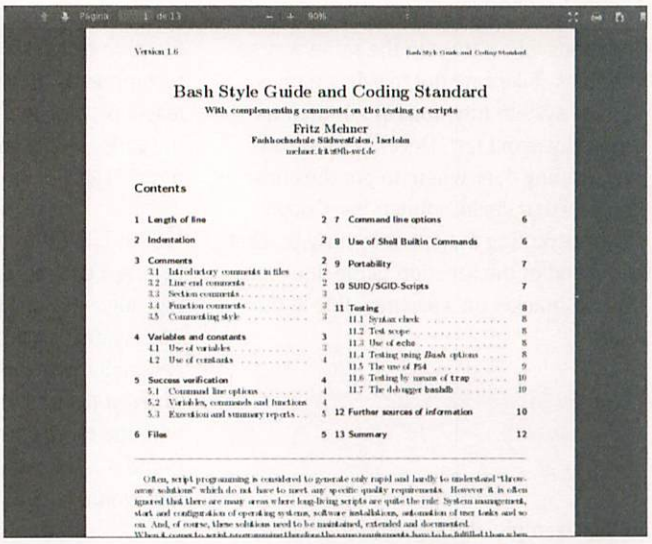


Figure 2: Fritz Mehner's booklet is handy, 13-page manual you can keep by your computer for reference.

done. As for *if* statements, Google recommends replacing `test` with double square brackets as abbreviations:

```
if [[ ${filename} == „image*" ]]; then ...
```

This notation has the advantage of allowing for the use of regular expressions. It also reduces errors in that the double square brackets neither arbitrarily truncate words nor expand filenames.

In the Break

In a *case* statement, list the alternatives indented by two spaces. The patterns and their values should each be on their own line. Listing 3 shows an example. Link multiple commands when necessary with a pipe, such as *ls -la | grep TODO*.

The pipe symbol should have a space characters on both sides. If the chain consists of more than two elements, it's best to enter each element on a separate line, with successive lines indented as shown below:

```
<command1> ➲
  | <command2> ➲
  | <command3>
```

Long parameter lists should follow the same principle:

```
./configure ➲
  --prefix=$HOME ➲
  --enable-audio ➲
  --enable-video
```

Furthermore, Google recommends wrapping lines at 80 characters, although other style guides allow for a slightly longer line length. These limits allow for easier printing and fewer scrolling hassles (Figure 3).

Storage Locations

If you name your variables with short, cryptic names, like `a`, what they represent can be deduced only in context, and then only with great difficulty.

Therefore, you should choose meaningful variable names that describe their contents. A variable named *filename* certainly can't be misinterpreted down the road. The names should be short and concise, consisting of no more than 31 characters. Take the pain of choosing suitable variable names, and it will pay off in the long run.

Most style guides suggest using lowercase variable names *(count)* and uppercase constants *(NUMBER)*. The constants should be defined as a group at the beginning of the script. Avoid using names that are reserved words for environmental variables.

You can find a list of these words on the web [3]. To be on the safe side, give constants a prefix, such as *ABC_COUNT*. If the name consists of multiple words, separate them with an underscore character, such as *letter_greeting*.

You're also advised to use singulars and plurals per context, which helps especially in loops:

```
for amount in ${prices}; do
  echo ${amount}
done
```

Always enclose variables in curly brackets, such as *${last_name}*. Enclose strings that include variables in quotes:

```
name="Mr. ${last_name}"
```

In that way, Bash correctly replaces the variable. With the new replacement names, the script in Listing 1 begins to gain clarity. Listing 4 shows a group of images that the script will process in certain ways.

Numbers appear in quotes only if they're part of a string. To ensure that false values don't enter calculations, always initialize variables. If the content of a variable shouldn't change, mark it explicitly with the keyword *readonly*:

```
result = do_something
readonly result
```

If a subsequent process changes the variable, Bash returns an error, which saves you a lengthy error search.

Listing 1: Unformatted Script

```
#!/bin/sh
a="100"; b="3"; c="."; d="shot"; for i in `seq 1 $a`; do import -window root $c/$d$i.png; sleep $b; done; exit
```

Nicely Embedded

Functions take names in the same way as variables. Take care not to select names that are system functions or commands – especially avoid *test*. Developers are forever arguing over where to put the curly brackets that enclose functions. Google suggests putting the opening curly bracket at the end of the function name line and the end bracket on a separate line at the end:

```
function my_function() {
<... commands ...>
}
```

In this example, the keyword *function* helps quickly identify the statement as such. However, this might complicate porting, so most style guides advise against using the *function* keyword. Identify local variables inside a function with *local*. This step helps avoid flooding the global namespace or possibly changing the wrong variable elsewhere:

```
calculate() {
  local amount=1
...
```

Get into the habit of checking the return value of a function. If a function returns nothing, you should at least confirm a successful end or an error. You must test all passed parameters. False or omitted parameters from users can lead to undesired results. If arguments are passed with *$@* or *$**, Google recommends always giving $@ precedence. Whereas *$** consolidates all arguments into one with a string value, *$@* leaves the arguments as they are [3].

Collect all function definitions at the beginning of the script, ideally just after the constants. To quickly find the beginning of a long script, wrap the main program in a function called `main`. Google's style guide recommends this for all scripts with at least one function defined. The very end of the script would then have the statement *main "$@"*.

Listing 2: Script with Proper Indentation

```
#!/bin/bash

a="100"
b="3"
c="."
d="shot"

for i in `seq 1 $a`
do
  import -window root $c/$d$i.png;
  sleep $b
done;

exit
```

Going Outside

Listing 4 uses so-called backticks (left-facing single quotes) to embed a command. This syntax is error prone and confusing, especially with embedded commands. Thus, you should replace the backticks with the more current notation *$(< command >)*. In Listing 4, the corresponding line would be replaced with the following:

```
for i in $(seq 1 ${NUMBER_IMAGES}); do
```

Always use full path names when processing a file, or your files might end up in the wrong directories. Using *rm* and other means to delete files also runs the risk of destroying needed directories.

Google gives as examples the dreaded *rm -v **, which deletes all subdirectories with it, and *rm -v ./**, which deletes all files on the hard drive. Also stay away from *ls* and its output. The returned filenames might contain line breaks, and different systems have different output conventions for this command. Also stay away from *eval*, which might give you loss of control over the code that the shell executes. Additionally, you can't tell whether the command was successful or how to change it if an error occurred.

Give the built-in Bash commands precedence and don't concatenate strings, for example, with external programs. For one thing, you don't know if the program even exists on other computers. For another, updates to the program can modify its behavior. Because Bash doesn't need to start a new process, running the script with the internal commands is much faster – especially if loops come into play.

If you're calling external programs in the script, use options in their long form, such as *grep --version* instead of *grep -V*. In this version, the meaning of the option becomes clearer in the longer form. If your script provides options, they should be available in long and short forms. Suggestions for developing parameters are included in the Free Software Foundation's GNU Coding Standards [5], and the most commonly used options are described at the Linux Documentation Project [6].

Picket Fences

Even if you're under time pressure, comment your scripts. After a week, you will still know what the script is doing and why. At the beginning of the script, briefly identify its purpose:

```
#!/bin/bash
#
# Captures screenshots
```

Some style guides recommend adding more, such as the author and version number. The comment hash mark (#) should be followed on the line with a space character before adding the comment text so that the comments are easier to locate.

Each function should have its own comments that briefly identify its purpose and methods, all passed arguments and returned values, and possibly which global values the function changes. Google recommends a structure like that shown in Listing 5.

If you're tempted to write a short novel as comments, your function is probably overly complex. You might want to think about splitting the function into smaller ones.

Finally, you should include comments for code blocks that might not be self-ex-

Listing 3: Sample case Statement

```
case „${auto}" in
  porsche)
    motor="hum"
    echo „${motor}"
    ;;
  *)
    echo „No known car"
    ;;
esac
```

Listing 4: Images for Processing

```
#!/bin/bash

NUMBER_IMAGES=100
WAITING_PERIOD_SECONDS=3
LOCATION="."
PREFIX_FILENAME="shot"

for i in `seq 1 ${NUMBER_IMAGES}`; do
  import -window root „${LOCATION}/${PREFIX_
FILENAME}${i}.png";
  sleep ${WAITING_PERIOD_SECONDS}
done;

exit
```

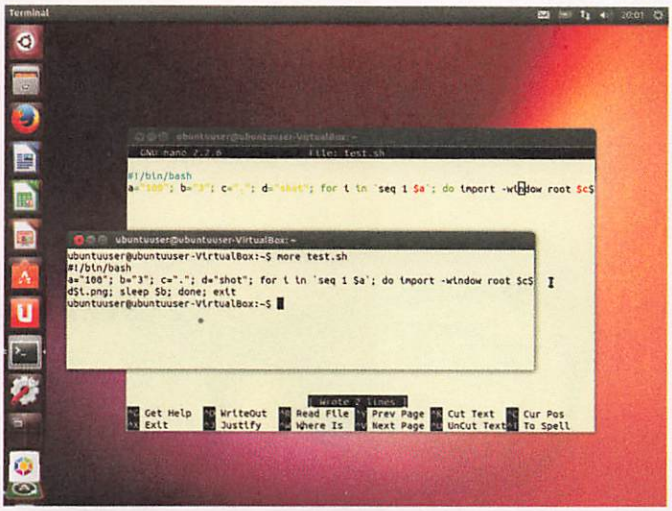


Figure 3: The terminal wraps lines at 80 characters. Many editors (e.g., nano in the background) truncate the line when the right edge is reached.

planatory. Good candidates are loops and if queries. Comments don't need to be too detailed, and you certainly don't need to comment each line. Some style guides suggest adding comments after the command line itself instead of on separate lines:

```
NUMBER=100 # False
# True
NUMBER=100
```

The comments should never just echo the code but simply describe its purpose. Instead of *for 1 through 100 calls image and waits 3 seconds*, write *captures a screenshot every 3 seconds*. Providing a good comment is often as hard as giving variables appropriate names.

If you've just quickly written the code with the intention of revising it soon after, you can add comments to target your further revision with a TODO, such as:

```
# TODO (tim@example.com)
# Calculate pseudo-random number
```

Listing 5: Recommended Structure

```
##########################################
# Captures screenshots
# Globals:
#   LOCATION
#   FILENAME
# Arguments:
#   None
# Returns:
#   None
##########################################
screenshot() {
  ...
}
```

In the parentheses give the contact person's name, ideally as an email address. Whoever sees the script later can then contact you. Some text editors even highlight lines beginning with *#TODO* so that they stand out. Apart from *TODO*, some style guides recommend adding *FIXME* for buggy sections or *XXX* for places needing particular attention.

Launch Pad

Remember to add the necessary rights to execute the script:

```
$ chmod +x <Script>.sh
```

Google advises omitting the *.sh* extension. None of the linked script files should have the execution bits set or retain the *.sh* in filenames. Filenames should all be lowercase. If you are concatenating words in filenames, separate them with underscores and not hyphens, such as *take_screenshots*. Remember not grant the script any *SUID* or *SGID* privileges to avoid risk of security breaches or misuse.

The script should provide users with termination information as to success or interrupted processing. In the latter case, it should provide a meaningful message on the STDERR channel. This approach allows all messages to be targetedly redirected later. End the script on error with *exit 1*, otherwise, with *exit 0*. That way the script can be more easily integrated into other projects.

Listing 6 shows the cryptic script in Listing 1 in its revised and improved form. If comments are in a foreign language, it's best to translate them into English. By the way, you can reduce the code in Listing 6 onto one line with a current version of ImageMagick:

```
$ import -window root ➥
  -snaps 100 -pause 3 shot.png
```

As you can see, referencing documentation or a man page can often save you from writing a script to begin with.

Conclusion

Some guidelines presented here may seem unnecessary. If you use them, however, you have a greater chance of writing clear Bash scripts that can easily be understood even at a much later date.

If you follow the guidelines, other programmers can work into your scripts more easily. For the sake of teamwork, agreeing on common scripting guidelines is a good idea. Additionally, adhering to these guidelines can reduce typos and errors. The standard you use is entirely up to you, as long as you stick with it. •

Listing 6: Revised and Improved Script

```
#!/bin/bash
#
# Captures screenshots

NUMBER_IMAGES=100
WAITING_PERIOD_SECONDS=3
LOCATION="."
PREFIX_FILENAME="shot"

# captures a screenshot every 3 seconds
# but not more than specified
for i in $(seq 1 ${NUMBER_IMAGES}); do
  import -window root „${LOCATION}/${PREFIX_
FILENAME}${i}.png";
  sleep ${WAITING_PERIOD_SECONDS}
done;

exit
```

Info

[1] Google Shell Style Guide: *http://google-styleguide.googlecode.com/svn/trunk/shell.xml*

[2] Bash Style Guide and Coding Standard: *http://lug.fh-swf.de/vim/vim-bash/StyleGuideShell.en.pdf*

[3] Environmental variables: *http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap08.html#tag_08*

[4] Special parameters: *http://www.gnu.org/software/bash/manual/bashref.html#*

[5] GNU coding standards: *http://www.gnu.org/prep/standards/*

[6] Standard command-line options: *http://tldp.org/LDP/abs/html/standard-options.html*

PlusServer stands for excellence and professional hosting:

- More than 30,000 servers worldwide
- More than 20,000 customers
- More than 15 years experience
- More than 30 different product types
- More than you would expect!

We offer the highest standard of hosting to you, so that you can provide your customers with nothing short of the best products on the market.
Our expert development, technical support, sales and customer service teams are at your disposal around the clock giving you the ability to be on top of the competition.

We offer everything you will need to start your own hosting brand – a control panel for each of your customers with lots of useful functions, the possibility to create your own design for the customer panel and to determine what features you want to offer them, and to make all sorts of other customizations.

Find out more:
reseller.plusserver.com

reseller.plusserver
PROFESSIONAL DEDICATED SERVERS

From simple queries to complex menus: Using dialog, you can create a graphical interface for shell scripts with only a few extra lines of code.

Create menus and dialogs for shell scripts

Greater Comfort

BY HARALD ZISLER

Christopher Hall, Fotolia

The shell by itself is already an application programming interface, but few users still query data from a command line. To adapt to the habits of current users, the dialog tool simulates the elements of a graphical interface.

The look may be a bit old-fashioned but, in terms of speed, the technology is hard to beat. If an X server is running on the system, you can provide users some additional comfort by relying on a dialog counterpart, such as Zenity or Gtkdialog (see the "Relatives" box).

Screens and Menus

To read user input, the built-in Bash *read* command is often used together with the *echo* command for simple strings. If you want to change the input prompt, enter the *-p* option with the new text (Listing 1).

The *-n* option on line 4 prevents a newline at the end of the *echo* statement, which makes the text appear directly to the right of the *read* prompt.

Quotes are escaped with backslashes so that they don't have any effect on the results. The two methods shown on line 4 and line 7 achieve the same results.

The option *-i <text>* together with *-e* (readline support) provides the user with an entry already at the prompt. This works a bit easier with the Readpreprompt [1] external program. For proper use, run the command in a subshell, as in Listing 2, line 3. The tool sends the result to standard output. This command can make it easier to create dialogs for database applications.

The built-in *echo* command has little effect on the outcome format. To accurately position numerical values in the dialog, the *printf* command borrowed from the C programming language can help. The command does the rounding for decimal numbers and

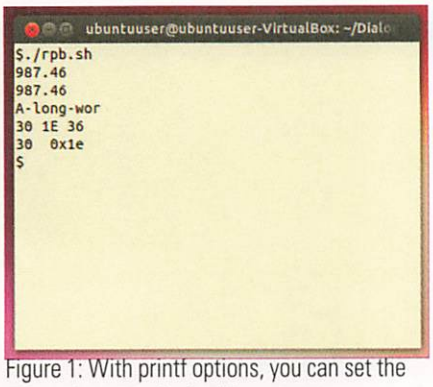


Figure 1: With printf options, you can set the output data to a desired format.

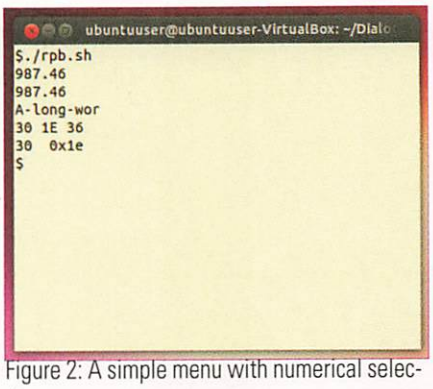


Figure 2: A simple menu with numerical selections.

takes the shell language settings into account.

The basic structure of the *printf* command is *printf "% < format > " < data >*. The most important instructions for formatting appear in Table 2. The command differentiates between period and comma as decimal separator. In a pinch, pass the variable *LC_NUMERIC* or *LANG* with *set* and *unset* inside the script.

An even easier method is to shoehorn output data into the desired format with *tr*, which you can add right in the statement (Listing 3, Example 2). Always end the *printf* statement with a newline (*\n*). In some cases, the command also allows tabbing. The examples in Listing 3 show the main functions, and Figure 1 shows the result.

Note especially the string output. A space character is typically a separator, which makes every word on the line a separate variable value. In Listing 3, you would get a five-line output if you used space characters instead of the hyphens to separate the words in the *c* variable.

When creating menus, proceed as in Listing 4. If there are just a few menu items, use numbers to identify them, which makes keyboard entry easier. If the numbers get too high, use lowercase characters. The script handles the function variables as strings, which avoids error messages when entries don't match. Figure 2 shows the example in action.

The example in Listing 5 (Figure 3) shows a dialog to process data for address management. For practical application, the functions for retrieving and saving the data would be from and to a database.

Relatives

Apart from the dialog framework described in this article, other projects exist with the same purpose. In Table 1, you can see to what extent other candidates provide dialog types. Some require a graphical user interface, among them Xdialog [3], Zenity [4], Kdialog [5], and Gtkdialog [6]. The latter deviates somewhat from the usual pattern: You create the XML files with the instructions the program reads in during the call.

Listing 1: Read User Input

```
01 #! /bin/sh
02 echo "--------------------------"
03 echo "echo -n \"Input: \";read a"
04 echo -n "Input: ";read a
05 echo "--------------------------"
06 echo "read -p \"Input: \" a"
07 read -p "Input: " a
08 echo "--------------------------"
```

With Dialog

The dialog [2] program comes preinstalled on many current distributions. The instructions consist of commands for designing the dialog, for defining the type of dialog (e.g., message, progress, input, etc.), and for geometric specifications. At the beginning of the instructions, you specify the title and background title. The commands also might require geometric settings.

The times of manually setting the size of the dialog are thankfully a thing of the past. If you want to omit the height in row and the width in characters, simply enter *0 0*, which works for most instructions. The tool then automatically adjusts the dialog proportions.

Listing 2: Using readprepromPT

```
01 #! /bin/sh
02 a="Old value"
03 a=$(readpreprompt "Input: " "$a")
04 echo $a
```

Table 1: SHC Options

Dialog Command	Xdialog	Zenity	Kdialog
--yesno	--yesno	--question	--yesno
--msgbox	--msgbox	--warning or --info	--msgbox
--infobox	--infobox	--	--passivepopup
--textbox	--textbox	--textinfo	--textbox
--tailbox	--tailbox	--	--
--pause	--	--	--
--gauge	--gauge	--progress	--progressbar
--form	--	--	--
--inputmenu	--2inputsbox/ --3inputsbox	--	--
--calendar	--calendar	--calendar	--calendar
--timebox	--timebox	--	--
--inputbox	--inputbox	--entry	--inputbox
--editbox	--editbox	--textinfo	--textinfobox
--dselect	--dselect	--file-selection	--getexistingdirectory
--fselect	--fselect	--file-selection	--getopenfilename/ --getsavefilename
--checklist	--checklist	--list	--checklist
--radiolist	--radiolist	--list	--radiolist
--menu	--menubox	--list	--menu

LISTING 3: FORMAT OUTPUT DATA

```
#! /bin/bash

# Sample values
a=987,455
b=987.455
c="A-long-word"
d=30

# Example 1
# Output floating-point rounded to 2 digits

printf "%5.2f\n" $b

# Example 2
# Output floating-point, comma converted to period decimal separator

printf "%5.2f\n" `echo $a | tr , .`

# Example 3
# Text output truncated to 10 characters

printf "%.10s\n" $c

# Example 4
# Numeric conversions at presentation:
# Integer, hexadecimal, octal
printf "%i %X %o\n" $d $d $d

# Example 5
# Output integer,
# Hexadecimal (lowercase) with leading "0x"

printf "%i  %#x\n" $d $d
```

LISTING 4: CREATE A MENU FOR USER INPUT

```
! /bin/sh
while true; do
  clear
  echo "(1) Function A"
  echo "(2) Function B"
  echo "(9) End"
  echo " "
  echo -n "Choose function: "; read f
  if [ "$f" = "1" ]; then
    echo "Function A";sleep 3
  elif [ "$f" = "2" ]; then
    echo "Function B";sleep 3
  elif [ "$f" = "9" ]; then
    exit
  fi
done
```

The following code shows the command structure.

```
$ dialog --title "Title" ➥
  --backtitle "Background Title" ➥
  <more parameters> 0 0
```

The type instructions indicate what the lines of code should do. Do you want the user to read, enter, or choose something? Table 3 shows the possible options.

In the example, the data lines are numbers and the functions are characters. The script refreshes the dialog after each change.

Although the yes/no and message boxes await a reply action, the info box is purely informational and will time out with no reply, based on the *sleep* or some other external setting.

You can expand the yes/no box with another button:

```
--extra-button --extra-label "<text>"
```

Clicking the button returns the value *3*. The second expansion possibility is the help button, *–help-button*, which doesn't require additional values and returns the value *2*. Thus, the yes/no box provides four possible return values.

The small script in Listing 6 shows the dialog settings in action and how to add properties. The properties in the first call are extended correspondingly. Before the *–yesno*, you'll find the additional parameter *–ok-label "<text>"*.

The same applies to the "No" button (*–no-label "<text>"*), "Yes" button (*–yes-label "<text>"*), and "Help" button (*–help-label "<text>"*). The *–defaultno* option in the last *dialog* statement sets the default selection to *No*.

For the gauge (progress) box, the percent value is piped to the dialog command (Listing 7). The element also displays units of an overall value.

Listing 8 is a "ready to use" script. It shows the current size of the */home* directory (Figure 4). Note this only works if */home* is on its own partition, and the extraction of values with *df* can be a bit confusing. The percentage appears as a status bar, and the free space shows as clear text.

LISTING 5: CREATE A DIALOG TO PROCESS DATA

```
#! /bin/sh
# Address sample data
# Database access would be enabled at this point
a="Mr."
b="Tux Penguin"
c="Icy Rd.  123"
d="South Pole 10001"

while true; do
  clear
  echo "----------------------------------------"
  echo "      Address Processing"
  echo "------+-----------+--------------------"
  echo "F-No. |           |    Value"
  echo "  1   |     Title: | "$a
  echo "  2   |      Name: | "$b
  echo "  3   |    Street: | "$c
  echo "  4   | City/Code: | "$d
  echo "------+-----------+--------------------"
  echo "actions: [F-No.]: change line, [s] save."
  echo "[q] quit"
  echo -n "action: ";read wn

  if [ "$wn" = "1" ]; then
    a=$(readpreprompt "Line $wn: " "$a")
  elif [ "$wn" = "2" ]; then
    b=$(readpreprompt "Line $wn: " "$b")
  elif [ "$wn" = "3" ]; then
    c=$(readpreprompt "Line $wn: " "$c")
  elif [ "$wn" = "4" ]; then
    d=$(readpreprompt "Line $wn: " "$d")
          elif [ "$wn" = "s" ]; then
    echo "This would be written to the database"
    break
  elif [ "$wn" = "q" ]; then
    exit
  fi
done

echo "----------------------------"
echo $a
echo $b
echo $c
echo $d
```

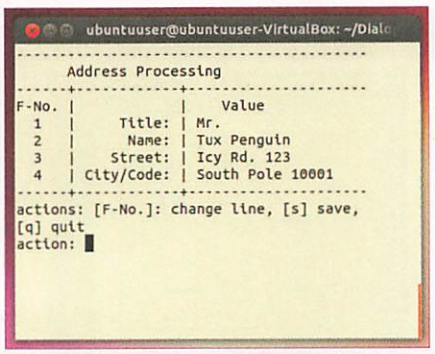

ubuntuuser@ubuntuuser-VirtualBox: ~/Dialo
Address Processing
F-No. | Value
1 | Title: | Mr.
2 | Name: | Tux Penguin
3 | Street: | Icy Rd. 123
4 | City/Code: | South Pole 10001
actions: [F-No.]: change line, [s] save,
[q] quit
action:

Figure 3: With just a few lines of shell code, you can create a dialog to process data from a database.

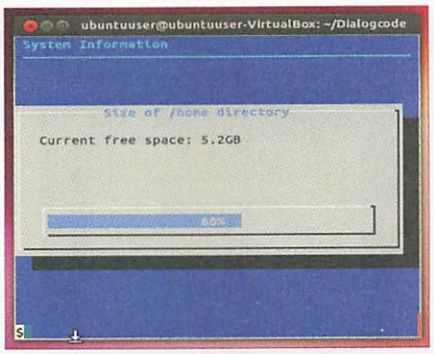


Figure 4: At a glance, you can see how full the /home partition is.

Table 2: Printf Instructions for Formatting

%5.2f	Floating-point with five digits before and two digits after the decimal separator
%.10s	String with a maximum 10 characters
%X\n	Hexadecimal in uppercase
%x\n	Hexadecimal in lowercase
%#X\n	Hexadecimal in uppercase with leading *0x*
%i\n	Integer
-r	Loosens the security settings during compilation, so that binaries can run on other computers with the same operating system. Currently mandatory for Arch Linux.
-D	Enables debug mode of the binary program, which can generate a lot of additional information.
-T	Creates a program traceable by Strace or similar tool.
-A	Shows a short info and quits SHC without compiling the script.

Forms

If you need not only simple forms but also a complex input box, dialog provides you with the necessary tools. You can navigate in the forms using the tab and arrow keys. The syntax might seem confusing at first, in that there are size and position adjustments to make.

Listing 9 is a script that allows editing of data and shows how to return it as variables that can stored permanently in a file. Figure 5 shows how the position and size values are rendered.

To ensure that redirecting output to a file works, you can precede *--form* with the additional *--stdout* option. In *msgbox*, use the newline character (*\n*) to output each variable value on a separate line.

Forms with the *inputmenu* function need fewer size and position parameters, but you may not get the data of all the fields at the end. You can change only single fields with each dialog call. The output includes the action (*RENAMED*), field identifier and data. To be sure the function works, use the *--stdout* option, as you do for *--inputmenu*. Post-processing of data occurs with *cut* (Listing 10). Here, it's important that the field names do not have space characters. Databases can provide the corresponding field names so that it's easy to provide SQL statements with matching variables.

The *--calendar* function provides an easy way to input date information. Calling it displays the current month and the day of the week in the left column. Use the cursor and up/down arrows to move to other data. Clicking the *OK* button returns the highlighted value.

To fill a variable, again use the *--stdout* option for output. Because dialog separates the data parts with a slash character, *tr* translates them into commas. Listing 11 shows how to construct a script that does just that.

You enter time values with *--timebox*. To accept the entered value, again use *--stdout*. The window shows the time the call to the function was made. To enter different values, use the arrow key to proceed to the next entry field and provide another date.

You can enter any alphanumeric values using the input box (*--inputbox*). Register a value in a script by using the prefix *--stdout* option. You can provide an editable default value. The process requires only one line:

```
a=$(dialog --title "Title" ➥
  --backtitle "Background Title" ➥
  --stdout --inputbox "HEADLINE" ➥
  0 0 "DEFAULT")
```

You can edit small text files in a mini-editor (*--editbox*). Give the filename as an argument. If no file exists, create one with *touch*.

You can write enter or modify text and then save to another file or overwrite the original file. Specify a height and width for the widget so that the inner window doesn't displace the headline. Again, use *--stdout* so that the entries don't end up in the standard error output. This function again requires a simple line of shell code, as follows:

Listing 6: Dialog Settings in Action

```
#!/bin/bash
#! /bin/sh
while true; do
  dialog --title "YES/NO BOX" --backtitle "BACKGROUND TITLE"
--help-button --extra-button --extra-label "EXTRA"  --ok-label
"Agree" --yesno "QUERY TEXT" 0 0
  dialog --title "MESSAGE BOX" --backtitle "BACKGROUND TITLE"
--msgbox "Return value: $?" 0 0
  dialog --title "END SCRIPT" --backtitle "BACKGROUND TITLE"
--defaultno --yesno "End shell script?"  0 0
  if [ $? -eq 0 ]; then
    exit
  fi
done
```

Listing 7: Gauge Progress

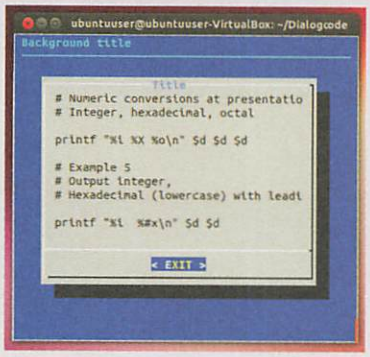
```
#! /bin/sh
percent=0

while [ $percent -lt 100 ]; do
  # Set the percent value
  percent=$(echo "$percent + 10" | bc)
  # Gauge value is passed via pipe to dialog
  echo $percent | dialog --title "GAUGE"
--backtitle "BACKGROUND TITLE" --gauge
"PROGRESS" 0 0
  # Sleep only for demo!
  sleep 1
done
```

DIALOG TYPES

Yes/No box

```
--yesno "<text>"
```

Yes: 0, No: 1



Tail box

```
--tailbox <filename> <height>
  <width>
```

Calendar box

```
--calendar "<text>" 0 0
```

Date

Message box

```
--msgbox "<text>"
```

0

Pause

```
--pause "<text>" <height>
  <width> <seconds>
```

-

Time box

```
--timebox "text" 0 0
```

Time

Info box

```
--infobox "<text>"
```

Gauge (progress) box

```
--gauge "<text>" 0 0
```

-

Input box

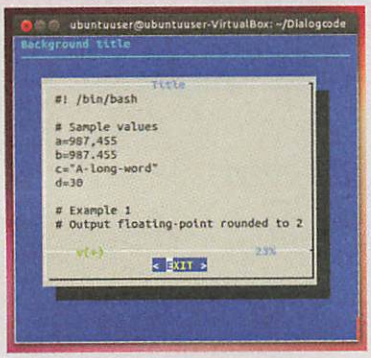
```
--inputbox "<text>" 0 0 "<de-
  fault-value>
```

Alphanumeric values

LISTING 8: SHOW CURRENT SIZE OF /HOME

```
#! /bin/sh

# Percentage derived from df -h

PERCENT=$(df -h | grep "/home" | tr -s ' ' | cut -d' ' -f5 | cut -d% -f1)

FREE=$(df -h | grep "/home" | tr -s ' ' | cut -d' ' -f4)

echo ${PERCENT} | dialog --title "Size of /home directory" --backtitle "System Information" --gauge "\n Current free space: ${FREE}B" 10 50

sleep 5
```

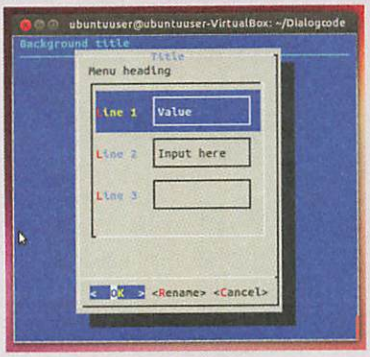
Dialog Types (II)



Text box

```
--textbox <filename> <height>
  <width>
```

0



Input box

```
--inputmenu "<text>" <height>
  <width> <menu-height>
  "<menu-item1>" "<value1>" ...
```

Data to standardout

Edit box

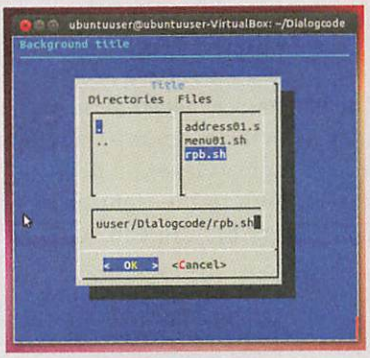
```
--editbox <input-file>
  <height> <width> > <out-
  put-file>
```

Text file

Directory select box

```
--dselect <directory> 0 0
```

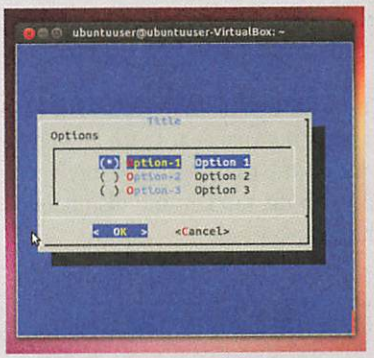
Directory name



File select box

```
--fselect <filename> <height>
  <width>
```

Filename with path



Radio button box

```
--radiolist "<text>" <height>
  <width> <number> <option>
  "<tag1>" "<text1>"
```

onoff

```
dialog --title "Title" --backtitle ➘
  "Background Title" --stdout ➘
  --ok-label "Save" ➘
  --editbox text.txt ➘
  20 75  > new.txt
```

You can select directories and filenames from within the script with *--dselect* and *--fselect*, respectively.

These two parameters are usually used together. It's usually a good idea to prompt users unfamiliar with the context. You select a directory at the same level as the default with arrow keys and pressing the space-bar.

Listing 9: Return Data as Variables

```
#! /bin/sh
a="Mr."
b="Tux"
c="Penguin"

dialog --title "Title" --backtitle "Customer Data" --ok-label "Save"
--stdout --form "Catalog" 10 60 3 "Salutation  " 1 1 "$a" 1 15 30 0
"Family Name " 2 1 "$b" 2 15 30 0 "First Name  " 3 1 "$c" 3 15 30 0 >
output.txt

a=$(cat output.txt | head -1)
b=$(cat output.txt | head -2 | tail -1)
c=$(cat output.txt | head -3 | tail -1)
rm output.txt

dialog --title "Title" --backtitle "Background Title" --msgbox "Saved
values: \n $a \n $b \n $c " 0 0
```

Listing 10: Post-Processing of Data

```
#! /bin/sh
newval=$(dialog --title "Title" --backtitle "Background Title"
--stdout --inputmenu "MENU HEADING" 17 60 15 "row-1 >" "Value 1"
"row-2 >" "Value 2" "row-3 >" "")
column=$(echo $newval | cut -d \> -f 1 | cut -b 9-)
entry=$(echo $newval | cut -d \> -f 2 | cut -b 2-)
dialog --title "Title" --backtitle "Background Title" --msgbox "Stored
Values: \n $newval \n $column \n $entry " 0 0
dialog --title "Title" --backtitle "Background Title" --msgbox "Saved
values: \n $a \n $b \n $c " 0 0
```

Listing 11: Translate Data with tr

```
#! /bin/sh
a=$(dialog --title "Title" --backtitle "Background Title" --stdout
--calendar "TEXT" 0 0 | tr \/ .)
dialog --title "Title" --backtitle "Background Title" --msgbox
"Selected date: $a " 0 0
```

Listing 13: Provide Multiple Selections

```
#!/bin/sh
a=$(dialog --title "Title" --backtitle "Background Title" --stdout
--checklist "SELECTION HEADING" 10 40 3 TAG-1 "INFO-1" on TAG-2
"INFO-2" off TAG-3 "INFO-3" on)
a=$(echo $a | tr -d \")
sleep 5
```

Listing 12: Select a Directory and File

```
#!/bin/sh
a=$(echo $HOME)
while true
do
  a=$(dialog --title "Title" --backtitle "Background Title" --stdout
--fselect $a 0 0)
  dialog --title "Title" --backtitle "Background Title" --defaultno
--yesno "$a Apply"  0 0
  if [ $? -eq 0 ]; then
    break
  fi
done
```

Listing 15: Add Menus

```
#!/bin/sh
while true; do
  a=`dialog --title "TITLE" --backtitle "BACKGROUND TITLE" --stdout
--menu "MENU HEADING" 0 0 0 1 "FIRST" 2 "SECOND" 9 "END"`
  if [ $a -eq 1 ]; then
    dialog --title "Title" --backtitle "Background Title" --msgbox
"First entry selected" 0 0
  elif [ $a -eq 2 ]; then
    dialog --title "Title" --backtitle "Background Title" --msgbox
"Second entry selected" 0 0
  elif [ $a -eq 9 ]; then
    dialog --title "Title" --backtitle "Background Title" --no-label
"End of program" --yes-label "Continue" --yesno "End of program?" 0 0
    if [ $? -eq 1 ]; then
      break # or exit
    fi
  fi
done
dialog --title "Title" --backtitle "Background Title" --msgbox "Saved
values: \n $a \n $b \n $c " 0 0
```

Choose the path on the left with *--fselect*; press the spacebar twice and choose the file on the right. Navigate between fields with the tab key and inside fields with the arrow keys.

The bottom field allows for manual entries. With the cursor in it, move to the next highest level by deleting everything up to the right-most slash (Figure 6). You can see how to select a directory and file in Listing 12.

Listing 14: Sample SQL Query

```
# Search dialog for the database query
sube=$(dialog --title "Customer Search" --backtitle "Customer Management" --stdout --inputbox
"Enter customer name" 0 0 "")
# Check for sufficient data
a=$(psql -t -c "select orgnr from customers where name = ,$sube';")
if [ -z "$a" ]; then
  dialog --title "Customer Search" --backtitle "Customer Management" --msgbox "No matching
customers found! " 0 0
  exit  # or break when in loop!
fi

# Get data for single-selection and select structure
# Counters for first "on" status
v=0
teil1=$(echo "a=\`dialog --title \"Customer Search\" --backtitle \"Customer Management\"
--stdout --radiolist \"found: \" 10 40 3 ")
# Determine customer numbers
for i in $(psql -t -c "select orgnr from customers where name = ,$sube' order by lastname,
firstname, birthday ;"); do
  if [ $v -eq 0 ]; then
    status="on"
    v=1
  else
    status="off"
  fi
  info=$(psql -t -c "select (lastname || , , || firstname || , , || birthday ) from customers
where orgnr = $i;";)
  part2=`echo $part2 $i \"$info\" $status`
done

selection_mask=$(echo "$part1 $part2\`")
echo $selection_mask
eval $selection_mask
dialog --title "Customer Search" --backtitle "Customer Management" --msgbox "Record number $a
selected " 0 0
```

You can provide multiple selections using *--checklist*. This option returns the designator of each entry enclosed in quotes. Field separators are space characters. You give each entry a status of *on* for marked or *off* for unmarked. You can specify space-separated values for height, width, and list height.

The list height should correspond to the actual number of items. If all the items don't fit, the box is scrollable.

Listing 13 shows the code together with *tr*, which removes the quotes. You can build a single-selection box in the same way by using the *--radiolist* dialog type and setting a single entry to *on*, getting the tag value without the quotes. Use the space bar to choose the value:

```
a=$(dialog --title "Title" ➥
  --backtitle "Background Title" --stdout ➥
  --radiolist "SELECTION HEADING" ➥
  10 40 3 TAG-1 "INFO-1" on ➥
  TAG-2 "INFO-2" off TAG-3 "INFO-3" off)
```

This changes the item selection to a single one. The script creates the data query dynamically. The tag consists of the record value or a unique value that the script uses to read in the record value.

The example in Listing 14 shows an SQL query with the psql client for the PostgreSQL database. The aim is to pass the customer number *orgnr* to other parts of the script. The script writes the instructions for the menu with *--radiolist* into a variable and

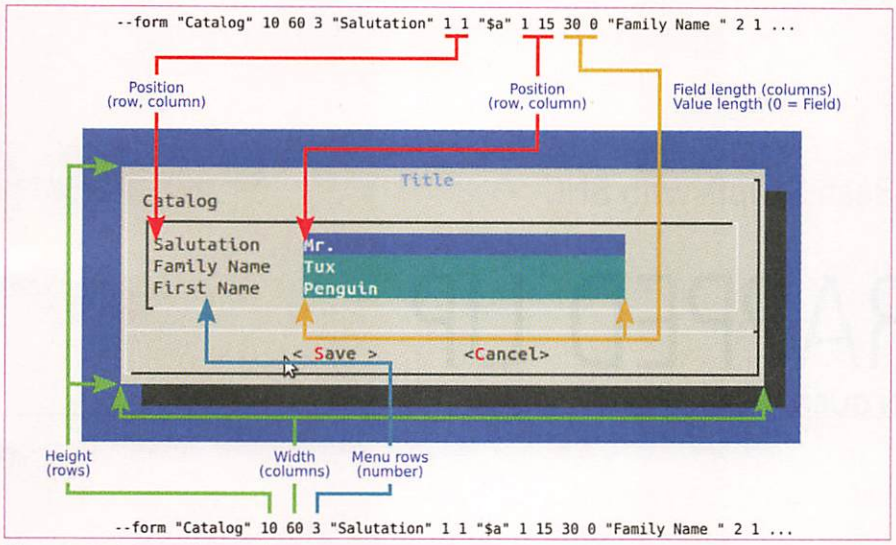

--form "Catalog" 10 60 3 "Salutation" 1 1 "$a" 1 15 30 0 "Family Name " 2 1 ...
Position (row, column)
Position (row, column)
Field length (columns) Value length (0 = Field)
Title
Catalog
Salutation Mr.
Family Name Tux
First Name Penguin
< Save >
<Cancel>
Height (rows)
Width (columns)
Menu rows (number)
--form "Catalog" 10 60 3 "Salutation" 1 1 "$a" 1 15 30 0 "Family Name " 2 1 ...

Figure 5: Position and size values for form elements.

executes the code with *eval*. For further development, the *–form* or *–inputmenu* dialog type can be used depending on programming preference.

You can add menus to your program with the *–menu* dialog type. It provides a tag that you later evaluate. As with almost all other value output, use *–stdout* before the dialog type. Listing 15 shows an example.

Conclusion

If you want to enhance your shell scripts with a user-friendly interface, you'll find everything you need with the dialog tool.

If you require simple queries only, you can benefit from prefabricated building blocks. Otherwise, you can always use something other than shell code. •

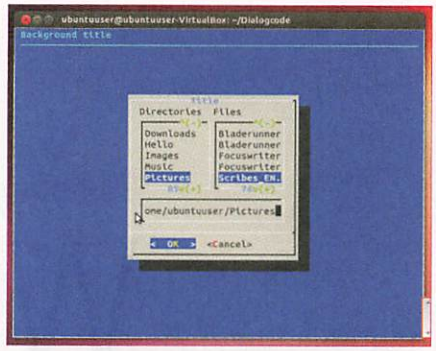


Figure 6: The --dselect and --fselect widgets make choosing directories and files easy.

Info

[1] Readpreprompt: *http://www.kev.pulo.com.au/readpreprompt/*

[2] Dialog: *http://invisible-island.net/dialog/*

[3] Xdialog: *http://xdialog.free.fr/*

[4] Zenity: *https://live.gnome.org/Zenity*

[5] Kdialog (KDE): *http://www.kde.org*

[6] Gtkdialog: *http://code.google.com/p/gtkdialog/*

Linux Magazine ACADEMY

20% off for current subscribers

Online Training with Linux Magazine Academy

Preparing for the LPIC exam - the easy way

Linux Professional Institute LPIC

GET YOUR LINUX KNOW-HOW CERTIFIED WITH LPIC:
Professional training for the exams LPI 101 and 102

- hardware settings
- package management
- partitioning and filesystems
- shell environment
- automate system administration
- network configuration
- security administration tasks
- troubleshooting

Equivalent to CompTIA Linux+ powered by LPI

For more information and to order: academy.linux-magazine.com/lpic

The Shell Script Compiler converts scripts into binaries, which protects against accidental changes but also carries some pitfalls.

Compiling Bash Scripts with SHC

WRAPPED UP

BY KARSTEN GUENTHER

Compiling programs makes sense for several good reasons. For example, you gain speed when running them. Additionally, compilers ideally generate small, portable programs that largely run independently of the computer on which they're created. Furthermore, this method protects the inherent algorithms from unintended changes.

bekas007, 123RF

The first two reasons – the gain in speed and portability – are not a focus of the Shell Script Compiler SHC [1]. The programs translated by SHC still require Bash as interpreter, so no great speed is gained. However, this point should not play a major role, because shell scripts aren't used for time-critical applications anyway.

Nevertheless, translating the source into binary code protects it from the ambitions of those users who want to make something good even better – while completely wrecking it. Currently, SHC is the most popular free tool for translating (Bash) shell scripts into fully functioning programs.

The operation of SHC provides some special features based on these reasons. Compilation occurs in two steps. SHC first generates from the script a fairly extensive, highly specialized C source code, then translates it into a binary program with help from the C compiler (see Figure 1).

In the first step, SHC generates a file with the *.x.c* extension, which it then translates into a file with the *.x* ending using the *$CC* C compiler environmental variable.

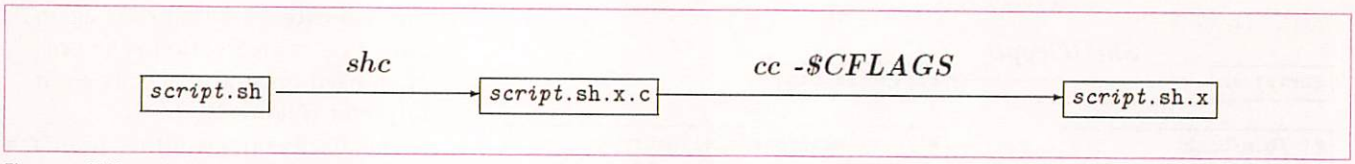


Figure 1: SHC works in two steps: It creates C code, then compiles it with the C compiler.

Listing 1: Calculate Correct Checksum

```
source=("http://www.datsi.fi.upm.es/~frosal/sources/${pkgname}-${pkgver}.tgz")
sha256sums=('\textbf{ef7bbf1252c9c791f711782870d00d6f19c42c08e0ee57e9a04d0e2b3d114d40}')
```

Converting the script source into C code done by using an array to capture all the script content. During compilation, SHC makes stepwise access to the (encrypted) elements of the array and integrates these entries into the executable program.

Details of the array and conversion to the binary program are comprehensively described in a blog [2]. The blog also covers the topic of password security in scripts. The author also describes ways to re-decrypt the programs created by SHC later on.

Installing SHC

Installing SHC is easy in Ubuntu and its variants because there are PPAs with more or less recent versions.

With Arch Linux, you have two obstacles to overcome. The current versions are in the Arch User Repository (AUR), but with the wrong checksum in the *PKGBUILD* file (Listing 1).

You can calculate the correct checksum by using the *sha256sums* tool and adding the corresponding query (Listing 2). The script opens an editor so that you can make the required change.

The SHC package also includes a series of test scripts (*pru.sh*, *test.bash*) that the package manager with Arch Linux doesn't include by default, even though the original archive does. These test scripts check that SHC functions properly, so you should generally use them to check the results.

Alternatively, you can generate SHC directly from the sources and install it. After unpacking the archive, a simple *make* is enough; *make install* installs the program in */usr/local/*. The target *make test* no longer works in version 3.8.9, so use *shc -f test.bash* instead.

In Practice

For a first test, use the classic "Hello World," although *echo "Hello SHC"* returns an alternative text. If the first script line doesn't contain *#! /bin/sh*, using the

```
shc -f hello.sh
```

command returns the message *shc: invalid first line in script:...*. Translation works with the shebang, and the *-v* option adds commenting (Listing 3).

The *hello.sh.x.c* source code thus generated is comparatively large at 9KB and largely incomprehensible at first, but it deals in large part with encrypting the script.

The executable program is not exactly small at 11KB and creates some problems on various platforms. Programs generated in Arch Linux and Ubuntu run on Arch Linux only when created with the *-T* option. This option actually allows observation of the program with external diagnostic tools such as Strace. Both systems run with no problem on Ubuntu.

Shell scripts have a few special properties that the compiler needs to understand and implement. This allows Bash to pass the script arguments that are position parameters inside the script. SHC can clearly understand them, even if you use *set* to reassign them.

The next important items that the compiler needs to check in shell scripts include return values ("exit codes") derived from internal and external commands.

Listing 2: Add Corresponding Entry

```
...
( Unsupported package: Potentially dangerous ! )
==> Edit PKGBUILD ? [Y/n] ("A" to abort)
==> ----------------------------------
\textbf{Y}...
```

Table 1: SHC Options

-e <date>	Limits the time the program can be executed up to the given date, after which a warning appears. The compiler expects the date in the form *<DD>/<MM>/<YYYY>*.
-m <message>	Message that appears after the *-e* time limit expires.
-f <script>	Absolutely necessary, indicates the script name.
-i <shell-options>	Special Bash options.
-x <command>	The binary program starts the script with *exec*, by default followed by *$@* (all command options and arguments).
-l <option>	Defines the last command line option, normally -- (the default).
-r	Loosens the security settings during compilation, so that binaries can run on other computers with the same operating system. Currently mandatory for Arch Linux.
-D	Enables debug mode of the binary program, which can generate a lot of additional information.
-T	Creates a program traceable by Strace or similar tool.
-A	Shows a short info and quits SHC without compiling the script.

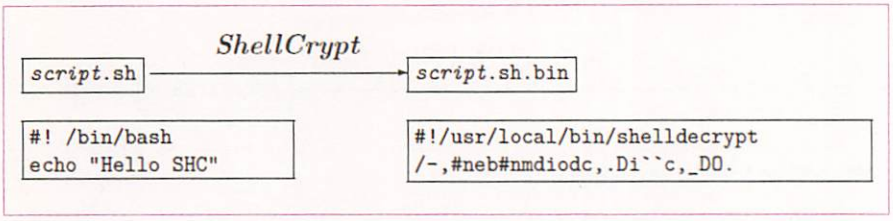


Figure 2: Shellcrypt converts scripts into unreadable characters that can only be made executable after back conversion.

With Bash, the *$?* variable contains this value for the last executed command in the foreground, and you read it with the *echo $?* prompt. SHC also supports this.

Particularly in shell scripts, return values occur in the context of linked commands. Bash provides some comfort with short-circuit tests: The double ampersand (&&) connects two commands, where the shell executes the second one only if the first one terminates without an error (exit code *0*).

Alternatively, you can use the double pipe (||). In this case, the shell executes the second command only if the first one terminates with an error (exit code not equal to *0*). Short-circuit tests are easily combined with SHC (Listing 4).

Another, sometimes touchy, subject involves inputs and outputs. Without additional software, the shell provides very limited options; however, these options always work. If you ask for input using Zenity or its new variant YAD, SHC does not get out of step:

```
INPUT=$(yad --entry "input" ➲
  --editable); yad --entry $INPUT
```

Here, the *INPUT* environment variable first gets its value through a YAD query. The input dialog is preset with the string *input*, which you can change because of the *--editable* option. The second call to YAD then shows the current content of the *$INPUT* variable.

When using external programs, most important to note is that SHC binds them into the binary program but calls them as in the script. This step assumes that the external programs are in the same path. The same goes for the calling Bash. Alternatively, SHC accepts absolute paths when running the compiled script.

Under Control

You control SHC mainly with options, the most important of which are shown in Table 1. Moreover, SHC evaluates two environment variables. *$CC* contains the C compiler used – default *cc*. The second variable in this context is *$CFLAGS*, which contains the compiler options.

Alternatives

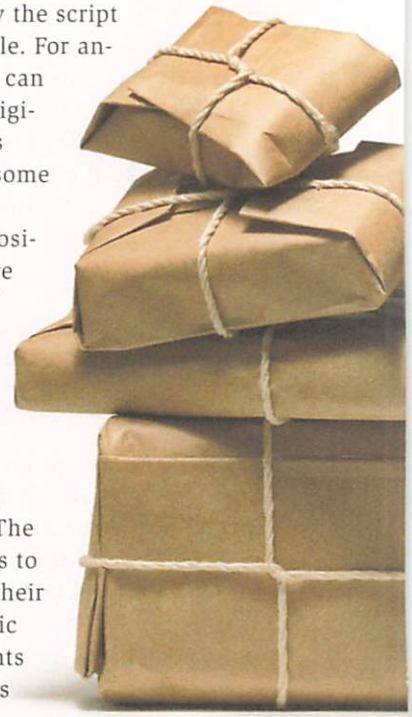
Several programs can help conceal the content of shell scripts. A search on Duckduckgo for *shell script encrypt* or *shell script obfuscate* returns some alternatives – often partly implemented as shell scripts – that convert code in one way or another. Among them are Obfsh [4] and Shellcrypt [5].

Although the former renders the source code unreadable by inserting or removing spaces and rows and other trash, Shellcrypt goes one step further. The program creates an truly encrypted program with the *.bin* extension that runs again only after re-decryption. The program used for it also acts as an interpreter (Figure 2).

The disadvantage of this approach is that the program also needs an installed Shellcrypt on the target system. In a similar way, you could use GPG encrypted scripts that you encrypt symmetrically with the *-c* option and re-decrypt with *-d*.

Conclusion

SHC is ranked somewhere between a toy and a useful tool. In no way can you count on the binaries created by the compiler to be truly encrypted. For one thing, they work only as long as the external programs required by the script are available. For another, they can read the original sources only with some difficulty.



On the positive side are the ease of use and the ability to prevent scripts from being modified. The alternatives to SHC have their own specific requirements or offer less portability. •

Listing 3: Test with "Hello World"

```
# shc -v -f hello.sh
shc shll=bash
shc [-i]=-c
shc [-x]=exec ,%s' "$@"
shc [-l]=
shc opts=
shc: cc  hello.sh.x.c -o hello.sh.x
shc: strip hello.sh.x
shc: chmod go-r hello.sh.x
```

Listing 4: Short-Circuit Test

```
$ true && echo "OK"   # OK
$ false && echo "OK"  # no output
$ true || echo "OK"   # no output
$ false || echo "OK"  # OK
$ true || echo "no" && echo "yes" # yes
$ true && echo "yes" || echo "no" # yes
```

Info

[1] SHC homepage: *http://www.datsi.fi.upm.es/~frosal/*

[2] Method: *http://www.l0d.org/Decrypt_an_shc_crypted_file-4.html*

[3] Retrieving text script: *http://stackoverflow.com/questions/3408373/retrieve-plain-text-script-from-compiled-bash-script*

[4] Obfsh: *http://www.comp.eonworks.com/scripts/obfuscate_shell_script-20011012.html*

[5] Shellcrypt: *https://sourceforge.net/projects/shellcrypt/*

Anyone doing off-road navigation needs special maps. You can create these maps in no time at all.

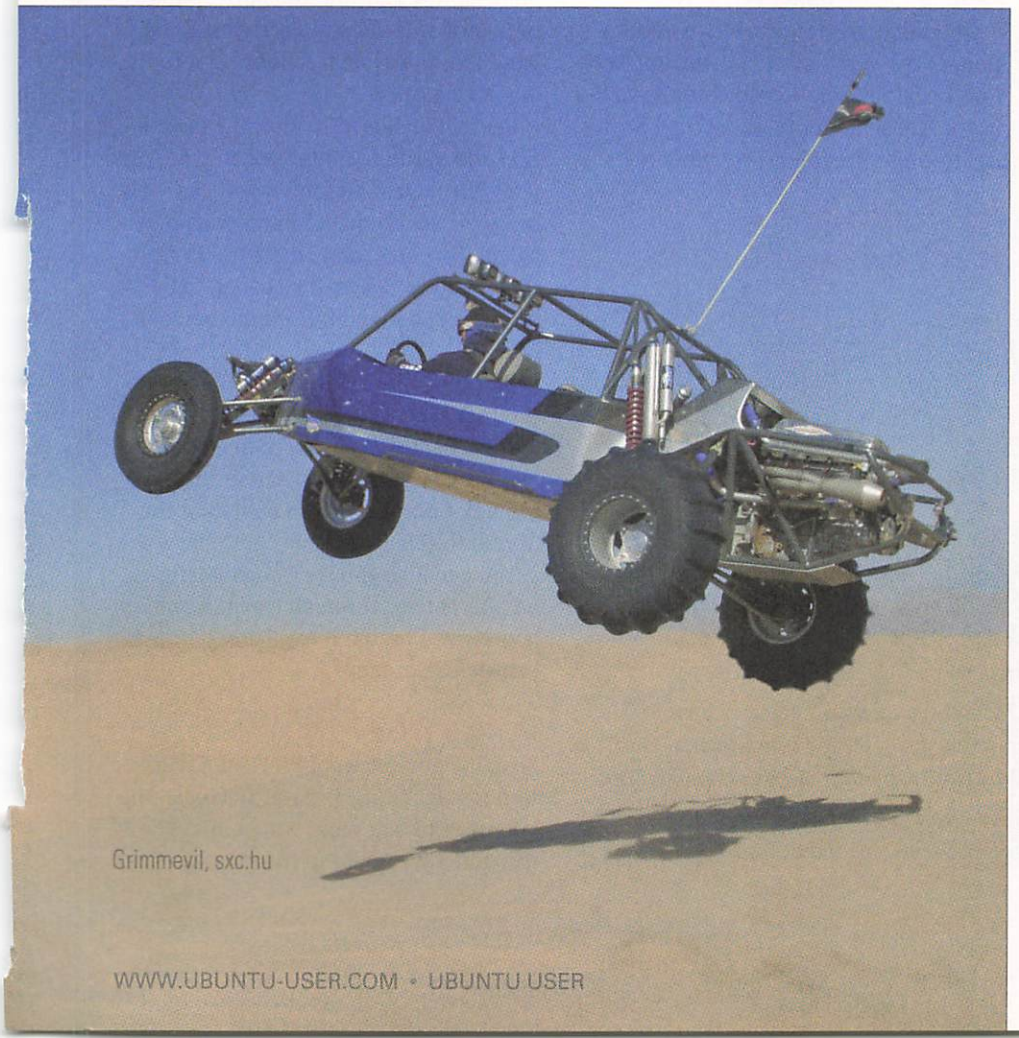
Software for DIY off-road navigation

In the Wild

BY KARL SARNOW

Moving a vehicle over unpaved roads requires a robust operating system to run the software. Linux qualifies as such, and Ubuntu is among the most popular.

In my example, Ubuntu runs on two computers: the home PC for preparing maps and the netbook that takes over the display and navigation per GPS receiver in your vehicle. The software to be installed consists of the following:



Grimmevil, sxc.hu

- QLandkarteGT [1] to test the maps
- Mkgmap and Splitter [4] to convert the map material from OSM format to the Garmin one
- Merkaartor [5] to create your own OSM maps

You only need to install QLandkarteGT on your netbook. The software together with the GPS receiver serve to display the maps, calculate routes and tracks, locate your position, and record the route.

Be warned: A lengthy off-road tour preparation has to take place on your home PC. The maps and tracks prepared on it before your trip get transferred via USB memory device to the netbook or a copied to disk for the software to read from.

Preparation

Installing Ubuntu on a PC usually presents no problem. Distributions other than Ubuntu usually do as well. What's important is using a current version of the aforementioned software for the distribution. For QLandkarteGT, use version 1.4.0 or later; for Mkgmap, you need version r2815 or later (with the corresponding Splitter at version r311); and Merkaartor requires at least version 0.17.2. Both Mkgmap and Splitter require Java version 7.

You can create cards directly from the OpenStreetMap server in a readable format (Garmin or GPX) for QLandkarteGT, although considerable computing resources are involved. For my tests, I used

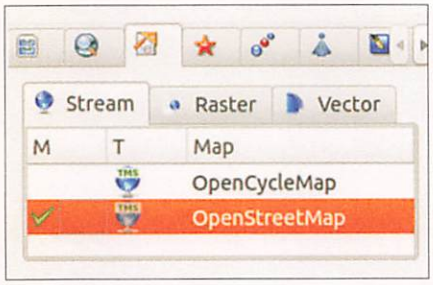


Figure 1: The OpenStreetMap server selection is in the toolbox under the Maps tab and the Stream subtab and forms the basis for the project.

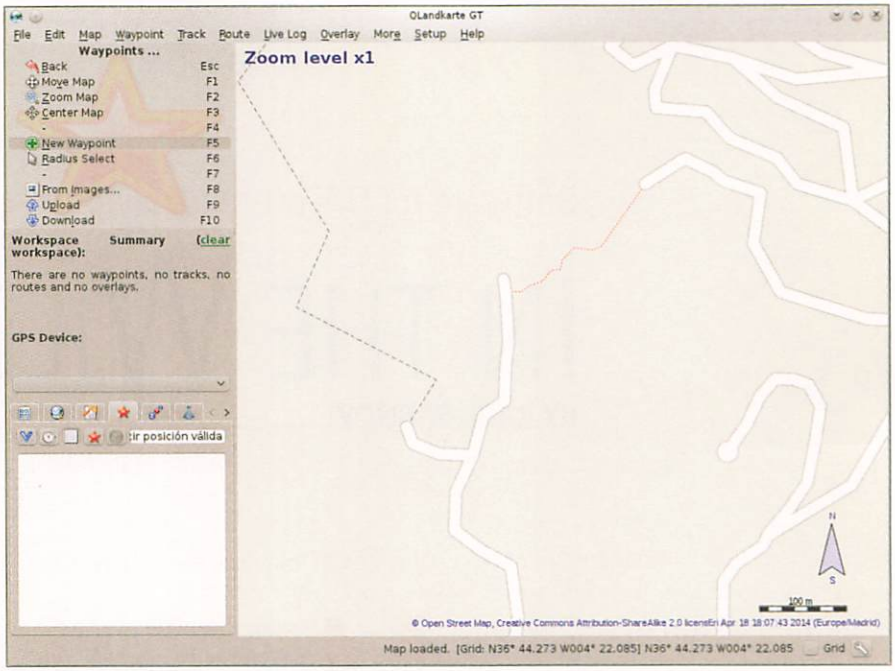


Figure 2: Select the Waypoints tab settings to create a new track point.

a computer with six CPU cores, 8GB RAM, and 21GB swap space. Even with such a powerful setup, the last successful attempt at creating a map took about three hours.

My attempt at converting an OSM map of Germany with Mkgmap into Garmin format crashed at the end of three days. Later in this article, I'll show how to make the conversion a success. For now, just be aware that processing maps requires good hardware and lots of patience.

If you've already decided at this point not to calculate your own maps, you won't need to install Mkgmap, Splitter, or Merkaartor. An easier way is to convert maps from OpenStreetMap to a format usable by QLandkarteGT for Garmin devices, which means that you need the programs GMapTool [2] and cGPSmapper [3].

QLandkarteGT

A central part of the project is the QLandkarteGT software. Although you can find it in most distribution repositories, including Ubuntu, it's usually woefully outdated.

With Ubuntu, you have the advantage of integrating already compiled newer versions from PPAs in the Software Center. You can then get the current repositories for installation instead of the outdated versions.

QLandkarteGT requires two additional repositories, one for the application itself and another for the GDAL library. These are both added in a terminal through the following commands.

```
$ sudo add-apt-repository ↵
  ppa:ubuntugis/ppa
```

```
$ sudo add-apt-repository ↵
  ppa:mms-prodeia/qlandkarte
```

Once you've integrated both repositories as such, the software list will be at its latest state.

The easiest way is through the package manager. First, load the list of programs anew and find the corresponding entries. If the software was already installed, the system will bring it to its latest version.

Start QLandkarteGT after the installation and verify that it's version 1.4.0 or later. With QLandkarteGT, you can view maps on the home PC, enter waypoints, calculate routes, and create tracks from them. You can save this geodata on a USB stick and read it from the netbook later on.

Working with QLandkarteGT requires Internet access, because the software accesses maps from OpenStreetMap. Select the OpenStreetMap server in the toolbox from the *Stream* subtab of the *Maps* tab (Figure 1).

You can also choose Google Maps as an alternative, although OpenStreetMap is preferred in that it explicitly permits private use of maps.

Setting Waypoints

A main component of off-road navigation is calculating a route. To create one you need two or more waypoints. You can add a waypoint to the map by click-

Listing 1: Splitting the File

```
01 $ /usr/lib/jvm/java-7-openjdk-amd64/bin/java -jar path-to-splitter-version/splitter.jar ./
   portugal-latest.osm
02 $ /usr/lib/jvm/java-7-openjdk-amd64/bin/java -jar path-to-mkgmap-version/mkgmap.jar --route
   --add-pois-to-areas --series-name="name-of-map-in-register"--index --tdbfile 6324.osm.pbf
```

Setting Up Splitter

Both Mkgmap and Splitter are Java programs that you can download from the Mkgmap project [8]. The project is quite active and the versions change often. Obtain both program as packed (TAR.BZ or ZIP) JAR archives that you can unpack in your favorite directory without further installation.

To determine whether the JRE is already installed and which version, look in your */usr/lib/jvm* directory. The directory contains the various Java versions. The one you need for Mkgmap and Splitter is the Java 7 OpenJDK for your architecture.

If */usr/lib/jvm* doesn't include a matching Java 7 version, install the OpenJDK Java 7 JRE either from the Ubuntu Software Center or directly from the Oracle [9].

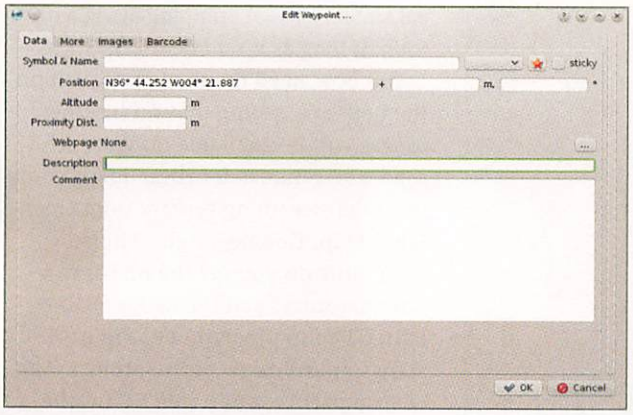


Figure 3: The waypoint dialog allows setting a permanent point. The coordinates are already indicated, with just a few more values to enter.

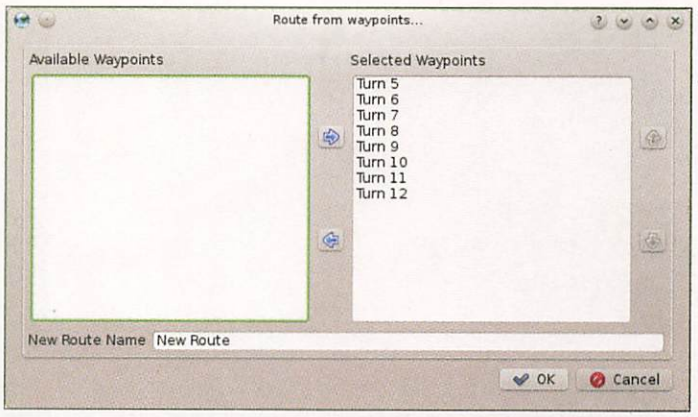


Figure 5: Use the horizontal arrows to remove waypoints from the route and use the vertical arrows to move the waypoints up and down to change the direction of the route.

ing the *Waypoints* tab (Figure 2) in the toolbox and then *New Waypoint* (or press F5). Then, click the map to create the point. Alternatively, you can click the *Waypoint* menu and the entry *New Waypoint*.

After creating the point a dialog opens (Figure 3) that already has the coordinates. All that's missing is a descriptive name, the symbol used on the map, and the option of making the waypoint permanent.

Set all the relevant waypoints before beginning your trip. Then, save the geodata in QLandkarteGT format by clicking the *File* menu and *Save Geo Data*. Be sure to select the *QLB (QLandkarteGT Binary format)* or else the application uses the GPX conversion format.

QLandkarteGT doesn't save the map itself, but rather the geodata contained therein – that is, waypoints, tracks, and routes. QLandkarteGT gets the map itself off the Internet from the OpenStreetMap streaming server. I'll cover creating a map on the USB storage media a bit later in this article.

Calculating Routes

Once you've entered the waypoints, you create a route from them. Note that QLandkarteGT creates a route according to the default from the open route server. For now, this calculates routes in Europe only – rather limiting for off-roaders seeking exotic corners of the world.

Destinations outside Europe therefore require an alternate route planner. MapQuest is one that is available for QLandkarteGT. The actual planning can then take place easily on the maps from OpenStreetMap or Google.

Figure 4 shows a route calculated from the OSM streaming map on the stored vector map of a Garmin device. Calculating routes on the home PC is extremely important for off-road use, because you rarely have Internet access on your netbook while on the road.

You create the route in three steps. First, mark all the waypoints that you want to evaluate with check marks. Second, right-click the waypoints and select *Create Route* from the context menu. The software then creates a direct route (straight line) between each waypoint. The program shows the selected points, and you can remove incorrect ones and give the route a name.

What's important is the direction of the waypoints. The software processes them from top to bottom; therefore, the starting point is the furthest up and the endpoint is the furthest down (Figure 5).

The third step is to use the routing server (on the *Routes* tab) to find the exact place on the map for the route. Do this by marking the polygon-shaped route and, with a right-click, choose *Calculate Route* from the context menu.

As a result, the software shows an icon on the route to indicate that it's being processed, then it loads the calculated route from the server. On the map, you'll see the route superimposed on the

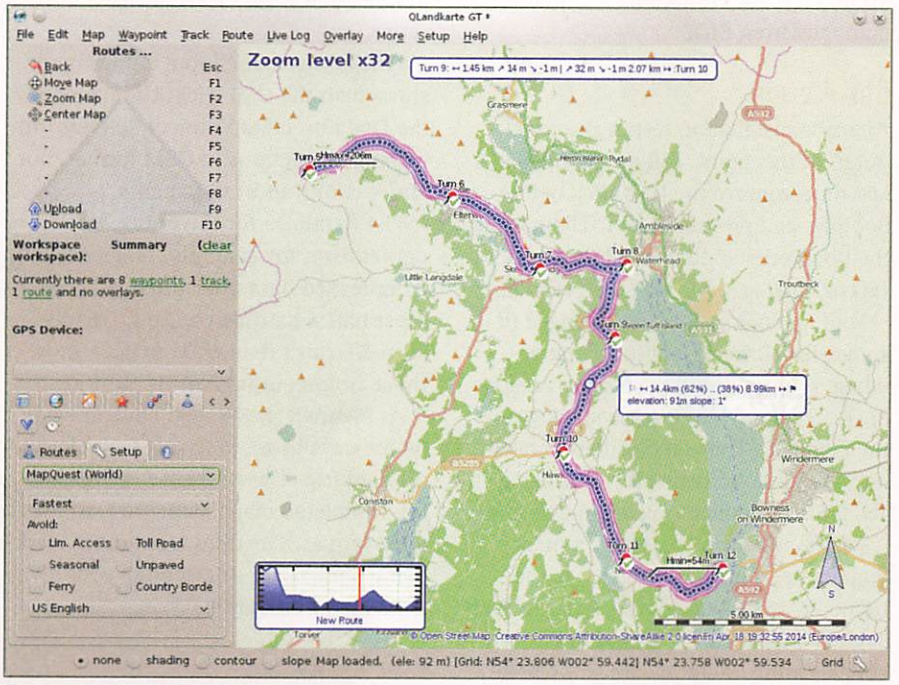


Figure 4: Although Open Route Service can only calculate routes for Europe, MapQuest can calculate routes for the whole world.

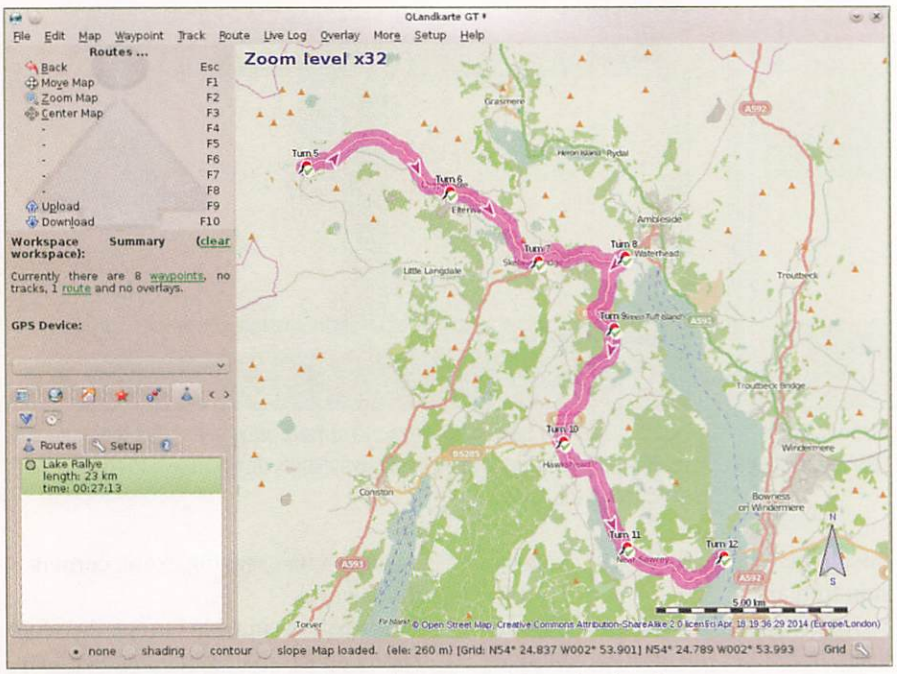


Figure 6: The finished route is matched to the paths and shows its direction.

road, with the specified direction indicated (Figure 6).

You need to use a robust netbook without any movable components, such as DVD players or hard drives, for the actual off-road trip – the netbook performance will end up being correspondingly slow. The installation is somewhat more complicated than that on the home PC.

Creating Boot Media

The Ubuntu homepage used to have a bootable USB image, but not anymore. Thus, you need a bootable CD or DVD with the current version. For the workshop, I used Ubuntu 12.04 LTS. If you already have installed Ubuntu on your home PC, use the *Startup Disk Creator* opened from the Start menu.

Otherwise, you can download the CD or DVD image or a bootable Ubuntu version and burn it to disk. Then, start the computer from the disk and create a bootable USB stick with the program. The USB stick should have a minimum capacity of 8GB.

The USB stick contains only the system for the actual installation to the SD card. If you no longer need the system on the built-in SSD hard drive, this can be a destination as well. In the case of the SD card, it's advisable to have a model with at least 16GB disk space.

Boot the EeePC and press F2 while starting it, which enters the system BIOS, where you select the USB stick as start media. Clicking the first entry has the BIOS show you the order of the listed drives. From this menu, be sure that the USB is at the top of the list. If not, choose the item and use the plus sign to move it to the top. Press F10 to save your settings, exit the menu, and reboot.

The system will now boot from the USB stick and provide Ubuntu 12.04 LTS, which you then install on the SD or SSD card. Indicate where you want to install GRUB, then you can start from the SD or SSD card without having to do so manually from BIOS.

Completion

After the installation restart the computer. At this point, call up the BIOS one last time, remove the USB stick before booting, and press F2 again. Then, reset the boot order so that the start media (SD or SSD) is at the top, then reboot.

Your netbook is now a full-fledged (if somewhat slower) Ubuntu computer where you can install the newest version of QLandkarteGT, as described above for the home PC. The netbook doesn't need any additional software. If you're using an 16GB SD card, there's plenty of free space.

All the work you've done on maps, waypoints, routes, and tracks on the home PC is best saved on a separate USB stick and used on the netbook. The resources on the netbook are still scarce. Using a USB stick shouldn't present any problems, even with respect to mechanical stability when bumping along a dirt trail, as long as it's a micro-USB stick.

You now need the maps installed on the netbook, seeing that no Internet access will be available on your off-road trip. Sources for these maps come from the streaming servers from OpenStreetMap, Google, or elsewhere.

So, how do you get the maps, waypoints, routes, and tracks for QLandkarteGT onto the netbook? Figure 8 shows an overview of the project. The central part is your home PC (at the left) with QLandkarteGT. On it, you create waypoints, routes, and tracks using the OpenStreetMap server and MapQuest route server.

Load the maps either in Garmin or OSM format and then convert them to QLandkarteGT format. Finally, save the maps, waypoints, routes, and tracks on a USB stick and read them into the netbook in your vehicle. You can then go on your way guided by GPS.

Waypoints, Routes, Tracks

As a basis, you indicate the waypoints for your map on your home PC in QLandkarteGT with help from your favorite map server. From your waypoints, you calculate the routes based on the Open Route Service (Europe only) or MapQuest. From these, you create tracks with altitude data with help from the Geonames.org server. These steps were described in my previous article in the series.

This is where the input comes in, or storage on the USB stick. Unfortunately, the OpenStreetMap project delivers only small map sections in OSM format, not entire countries or continents.

The database for the entire world, *planet.osm*, for example, takes 32GB of compressed data after unpacking, almost half a hard drive. And, QLandkarteGT can't read OSM maps – only those in uncompressed Garmin format.

As Figure 8 shows, you have three different ways to get to usable maps for the off-road trip. Two of them require loading completed maps, either in Garmin or OSM format. The third takes more work, using Merkaartor to assemble your own OSM map from parts.

Free Maps in Garmin Format

To avoid misunderstanding, the term "Garmin maps" in this article refers

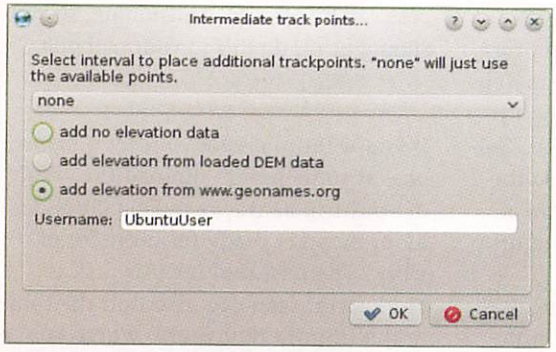


Figure 7: If possible, the tool integrates information about the altitude from the geonames.org server when calculating tracks from a route.

to free OSM maps in Garmin format. Inquisitive users have already partially reverse-engineered the undocumented Garmin format [6], as described in some detail on the project website on SourceForge [7]. Volunteers are calculating the format from OpenStreetMap sources and providing them for download, thus the material is neither produced nor supported by Garmin itself.

The maps are actually from the OpenStreetMap project, but in Garmin format, and the authors explicitly allow free use of them. Please read the authors' terms of use – some request payment or apply certain restrictions based on payment.

Editing these maps isn't recommended, because it is a hard and tedious task (more on this later). If you still want to modify them based on errors you detect, use Merkaartor together with the OpenStreetMap server (also more on this later).

Your changes will hopefully make it back sometime to the OpenStreetMap map in Garmin format. Critical handling of OpenStreetMap maps is especially needed if you want to traverse unknown or particularly dangerous territory. Additionally, you should bring along a good map from a well-known geographic publishing house and a compass.

The free Garmin format maps are distributed across multiple websites. The OpenStreetMap wiki [8] provides a comprehensive list of these sites. You usually download them as compressed files, with sizes in the gigabyte range, based on the region, country, and continent. Not only does the download take a while, but the unpacking does, too: In my test, a *OSM_egeneric* took the AMD six-core processor with 8GB RAM almost two minutes to unpack.

Different Archives

Unpacking the map data results in either a single IMG file or a directory with a TDB file and an IMG file of the same name. The latter case is a bit easier: No conversion is necessary, and you can read the map directly into QLandkarteGT.

To do so, click the *File* menu in QLandkarteGT, then *Load Map*. Next, open the TDB file and immediately afterward the IMG file of the same name. Both bring up the new map that will appear on the *Maps* tab under *Vector*. Based on your zoom level, you might see only a white area; by zooming in, you'll eventually see the map details.

Some archives have multiple versions of the TDB files. Then, it's only a matter of finding out which one works in QLandkarteGT without crashing. Now's a good time to get it right, because you'll have other work to do on your trip.

Saving the directory with all the files on a USB stick lets you load them on your travel netbook. The USB stick then must be plugged in all the time, or you can copy the directory onto the netbook's SSD.

If the map is a single IMG file, you'll need to extract the individual grids (as IMG files) and the TDD file from it.

To do so, copy the programs GMapTool and cGPSmapper-static in the directory and start a terminal window. In it, change to the directory where the map and both programs are and execute the following commands:

```
$ ./gmt -S <file>.img
$ ./cgpsmapper-static mapset.mp
```

For *file*, use the name of the map file that you get after unpacking; *mapset.mp* is a fixed name.

Whereas GMapTool chops up the IMG into grids, cGPSmapper-static

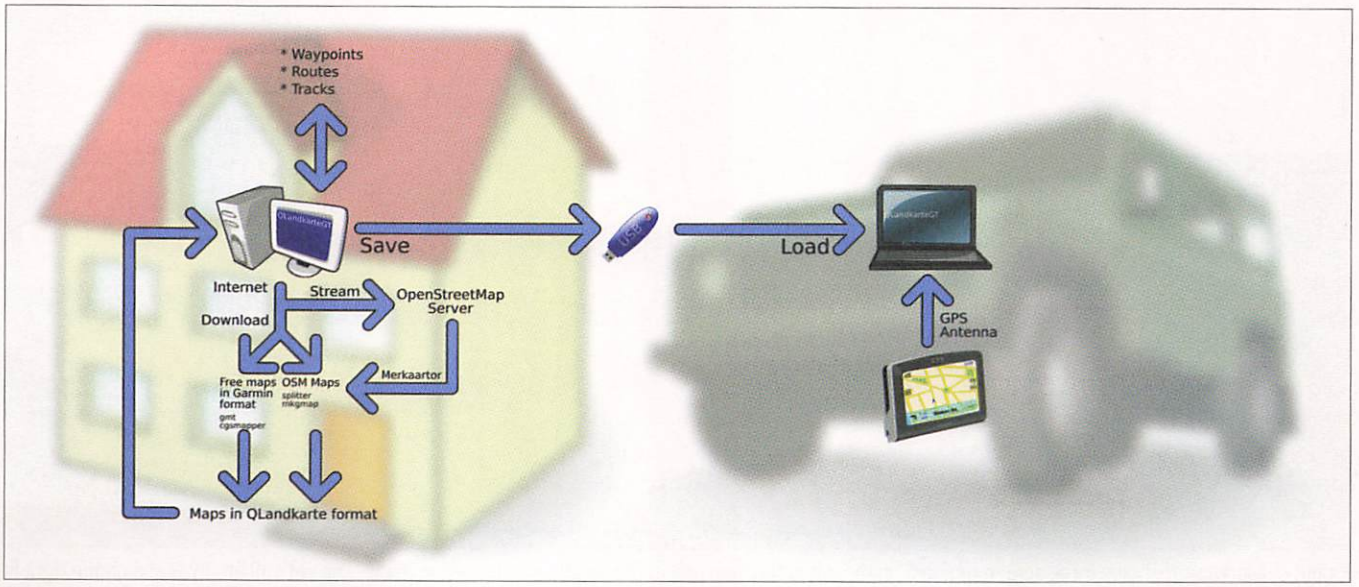


Figure 8: An overview of the entire navigation project.

creates the *mapset.img* and *mapset.tdb* files, which you then load with QLandkarteGT as described earlier. Again, you copy the whole directory to the USB stick to load it on the netbook.

Downloading OSM Maps

A few sources provide ready-made maps in OSM format – the original format of the OpenStreetMap project. An especially good source is the GeoFabrik server that includes numerous OSM maps that are updated daily.

I tested the reasonably-sized OSM map of Portugal. The corresponding *portugal-latest.osm.bz2* file had a size of 110MB after downloading. Upon decompression, the *portugal-latest.osm* file took up about 1.5GB space.

The first attempt at converting the map with Mkgmap into a format readable by QLandkarteGT returned an error that the map was too big and, therefore, needed to be broken up.

The Mkgmap homepage recommends the Splitter [10] program for this purpose and provides the matching download link. After setting up the small Java program (see the "Setting up Splitter" box) change to the OSM map's directory. There you enter the command for splitting up the file (Listing 1, line 1).

The *portugal-latest.osm* Portugal map took a good 11 minutes to process. The map directory is then filled with many map segments that you process further with Mkgmap. Next, execute the second command on the next line in Listing 1.

The *--tdbfile* option serves to read the resulting map directly into QLandkarteGT. The *--series-name* option specifies the name under which the map should appear in QLandkarteGT's map register.

In the Portugal example, I had data collection readable by QLandkarteGT after about three and a half minutes. Even with a comparatively smaller map like Portugal's the map preparation process can be resource-consuming, so you should never do this in your netbook.

Creating a Few Maps

If you want to create your own maps or – at some point – upload the result of your trip as tracks to OpenStreetMap, then get ready for quite lot of work. The best tool for this task is Merkaartor. You can also find the tool in the Ubuntu repository, where you can easily access it through your package manager.

Merkaartor proves to be a rather resource-hungry program. The minimum RAM required is 8GB and significant swap space for larger maps. A faster processor would reduce the waiting time during processing, and a wide monitor would help because of the program's window distribution.

Starting Merkaartor

After starting, Merkaartor shows three screen regions. On the left is the (still empty) selection of loaded maps. In the middle is a pixelated area that will even-

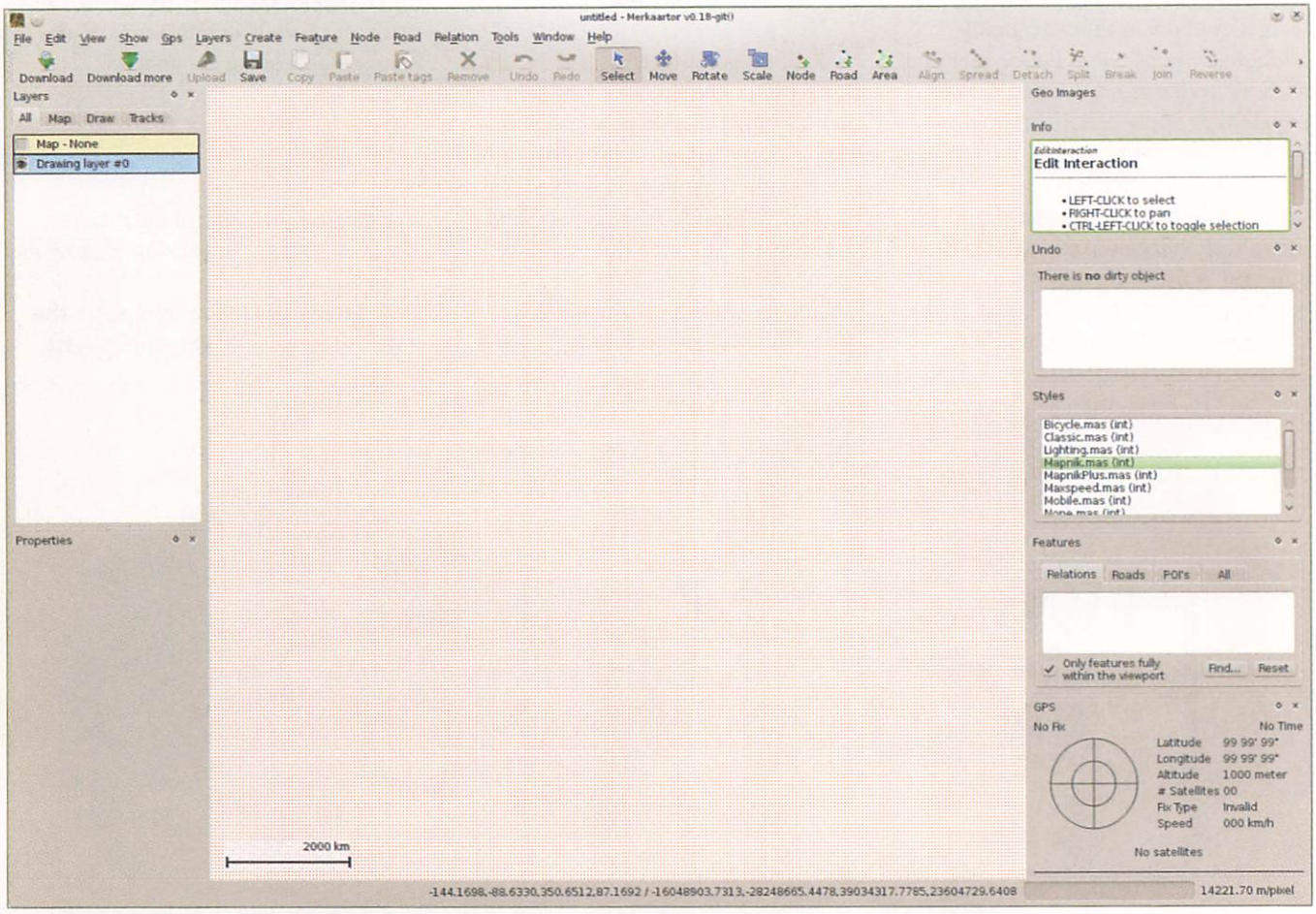
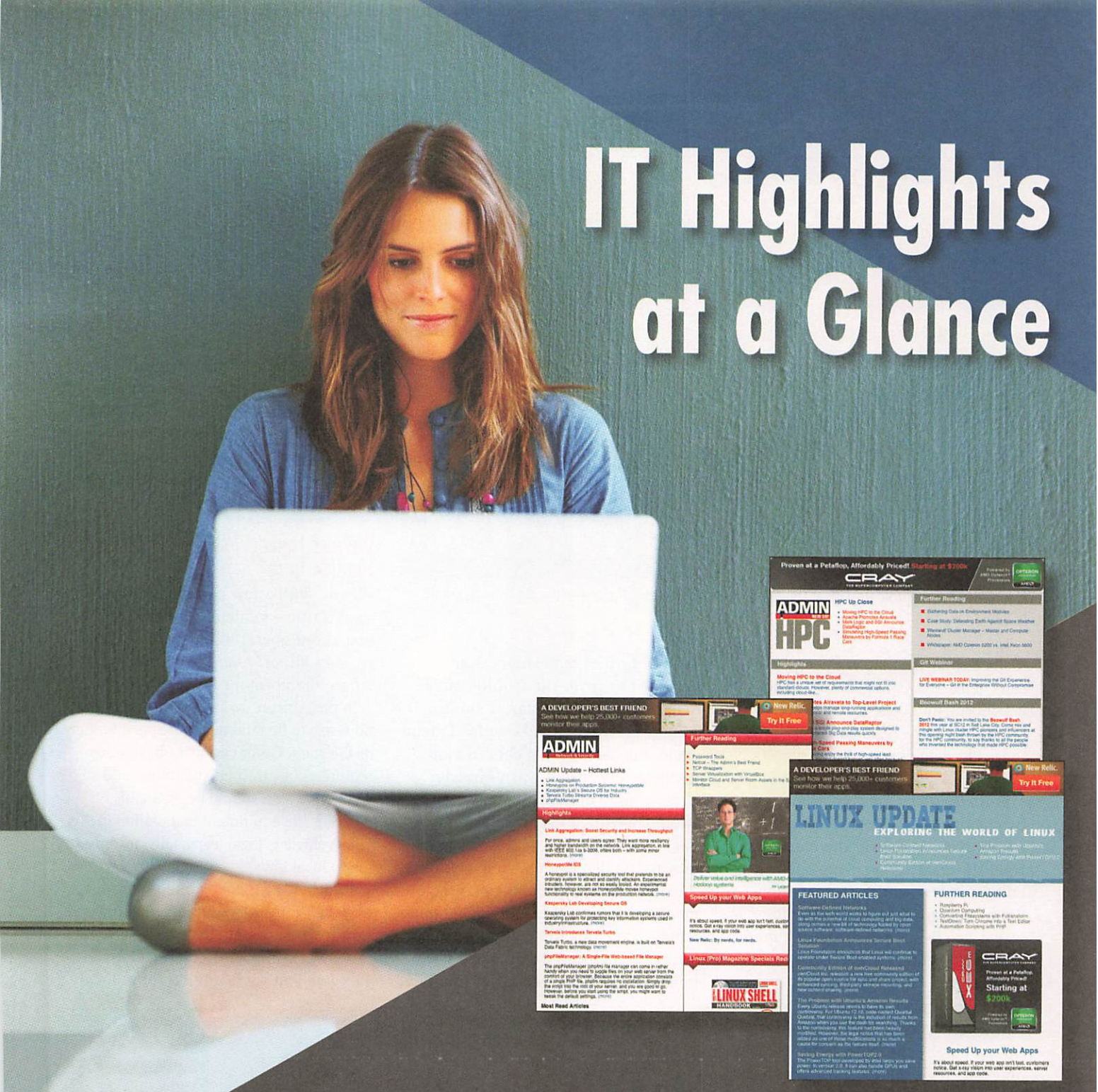


Figure 9: Merkaartor shows three screen regions: on the left is the map list, in the middle are the actual maps – as yet still empty – and on the right is the status of the selected map details.

IT Highlights at a Glance



Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your in box. Subscribe today for our excellent newsletters:

- Linux Update
- ADMIN Update
- ADMIN HPC

and keep your finger on the pulse of the IT industry.

Admin and HPC: www.admin-magazine.com/newsletter

Linux Update: www.linuxpromagazine.com/mc/subscribe

©Yuri Arcurs, 123RF.com

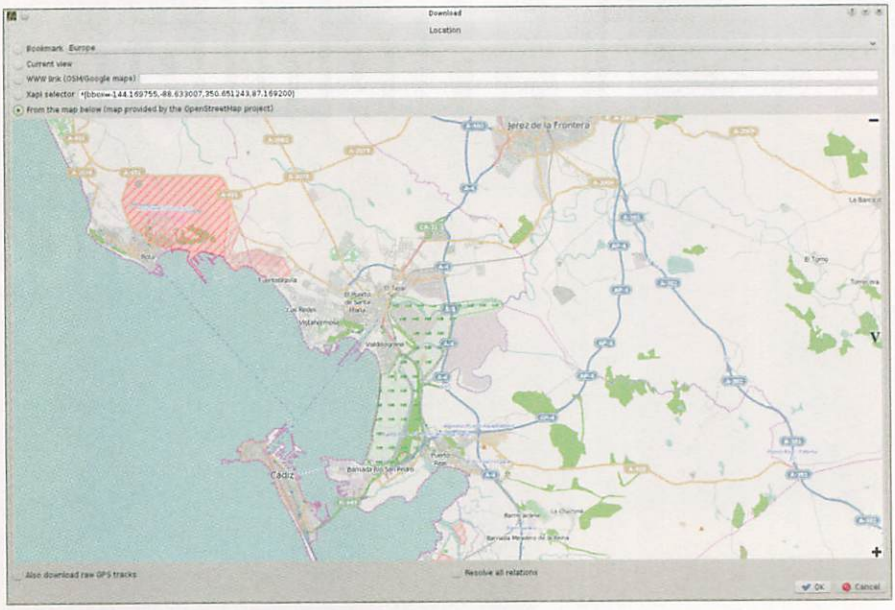


Figure 10: Loading from the OpenStreetMap server is probably the easiest using From the map below....

tually display the map segments. You can change this configuration, although it's outside the scope of this article to explain how. In the region on the right is the status data of the selected objects (Figure 9).

From the *File* | *Download* menu (or by clicking the button by the same name), you download the map data from the OSM server. This opens an entry window where you can highlight the map regions in different ways. The best one is *From the map below...* (Figure 10).

When you click *OK*, Merkaartor starts the download. At first, you may end up choosing a too large map section, whereby the OSM server returns an error. Try zooming in on the map a bit more until the OSM server stops complaining. When you're done, the middle region of the Merkaartor window displays a lot of lines and points – that is, the raw OSM map data (Figure 11). The map becomes more readable as soon as you apply a background. Right-clicking the *Map* item on the left provides a selection of backgrounds (Figure 12).

Clicking the box in front of the map field displays the background and makes it easier to interpret the data. The background, however, is not part of the stored map.However, you're still short of a usable map at this point. The map segment is far too small for effective Merkaartor processing and further map snippets need to be added to cover the required area.

This is where Merkaartor shines. When you call up the *Download* menu again, you see the last chosen map segment. Because it wasn't overly large, simply moving the map over to another segment yields an extended one of reasonable size. Repeating the process (which can take a few minutes) eventually leads to a map segment that covers the entire geographic region. Save this file in Merkaartor format, which can again take a few minutes and even an hour or more for a large map. The process is also heavy on memory and CPU usage.

Saving OSM Files

Although Merkaartor saves files in its own format, to be able to work with it outside of Merkaartor, you will have to export it to OSM at some point. This process can take a long time, perhaps an hour or more. I recommend converting the files, as described earlier, into a pair of TDB and IMG files that QLandkarteGT can read. Figure 7 shows the resulting conversion loaded into QLandkarteGT for the map in Figure 13.

Vehicle Inspection and Traffic Regulations

I'll end with a few comments about road traffic – vehicular traffic isn't just about operating the vehicle. This is, after all, *off-road* navigation – where the normal rules don't apply.

In the African savannah, there's really no sense to be talking about speed limits or the such. Other countries have other requirements, traffic regulations, and so on. Hence, you should be aware of whether the equipment installation in your vehicle is legal, especially according to local laws.

To provide at least some assurance, be sure that the equipment installation in your vehicle is done by a reputable workshop. Automobile clubs can give you some advice about local

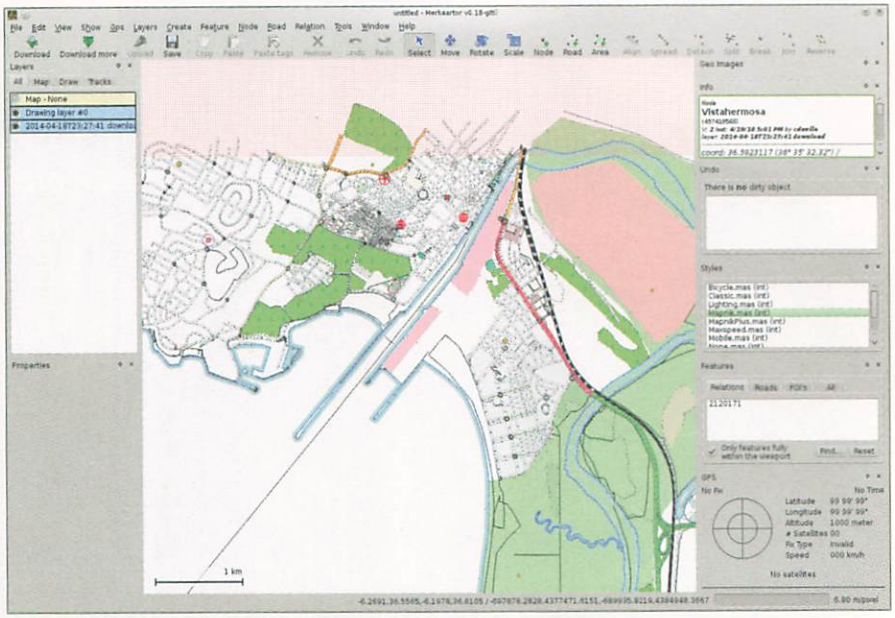


Figure 11: The raw OSM map data can be cryptic. Easily recognizable, however, is the downloaded map region (the little red points).

Figure 12: The map background can be from various server data. The one shown is from OSM Mapnik.

regulations. For example, some netbook mounts can be easily installed and uninstalled, which can lead to trouble in some countries where this isn't allowed, and you might have to go without off-road navigation.

In any case, make sure that the vehicle's security functions aren't inhibited by the mounts. A netbook in front of an airbag wouldn't work well in an accident.

In daily use, the usual on-road navigation systems are superior to the off-road ones in this article. They can talk to the driver and allow hands on the steering wheel.

These DIY navigators don't have this capability. Therefore, a passenger should be readily on hand to check the screen and the surrounding landscape while the driver pays attention to the driving task. On-road and off-road are very different scenarios.

Conclusion

The first part of this series dealt with the required hardware on the basis of an EeePC. It's left to your imagination how you adapt your own old hardware correspondingly. You can also pick a distribution different from Ubuntu 12.04 LTS.

In this second and final part of the series, I have presented the necessary software such as QLandkarteGT, Mkgmap, Merkaartor, GMapTool, and cGPSmapper.

Now you have all the tools and knowledge at hand to obtain OSM maps from the Internet to create your off-road tour and traverse the landscape over waypoints, routes, and tracks. Creating a map with Merkaartor is time-consuming and is only necessary when no other usable maps for a region are available.

With all the trust in your netbook and reliability of Ubuntu, you should keep two further things on hand in the wilderness: a good magnetic compass and a detailed printed map of the region. Neither a compass nor a geological survey map require a reboot – and both keep working when the battery on your computer is dead. •

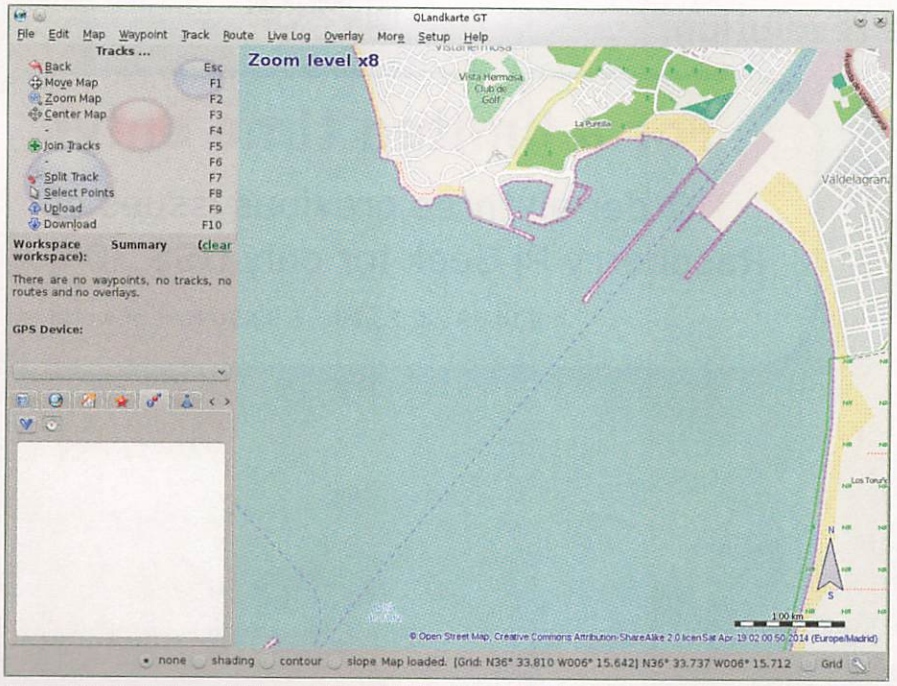


Figure 13: Worth the wait: The outskirts of Cadiz downloaded by Merkaartor from the OSM server, saved as an OSM file, and made readable by Splitter and Mkgmap for QLandkarteGT.

Info

[1] QLandkarteGT: *http://www.qlandkarte.org*

[2] GMapTool: *http://http://www.gmaptool.eu/en/content/linux-version*

[3] cGPSmapper: *http://cgpsmapper.com/buy.htm*

[4] Mkgmap and Splitter: *http://www.mkgmap.org.uk*

[5] Merkaartor: *http://wiki.openstreetmap.org/wiki/Merkaartor*

[6] Garmin map format: *http://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin/Format*

[7] Reverse-engineering Garmin maps: *http://sourceforge.net/projects/garmin-img/*

[8] Downloading OSM maps as Garmin: *http://wiki.openstreetmap.org/wiki/OSM_Map_On_Garmin/Download*

[9] Maps in OSM format: *http://download.geofabrik.de*

[10] Introduction to Mkgmap: *http://wiki.openstreetmap.org/wiki/Mkgmap*

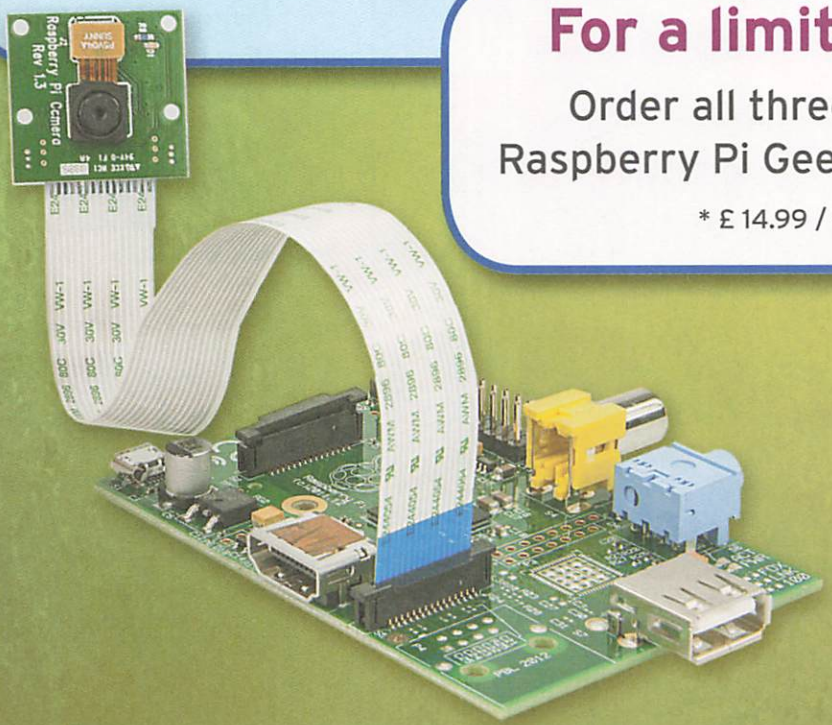
[11] Oracle Java: *http://www.java.com/download/*

Shop the Shop shop.linuxnewmedia.com

Make your Pi run faster
We can show you how!
raspberry pi GEEK
TUNE YOUR PI
Yes you can make it run faster
Rasp Pi as a wireless remote device
Practical Pi Set up a wireless print se
Gardening with Arduino
Free DVD!
Best Rasp Pi distros ever
WWW.RASPBERRY-PI-GEEK.COM
New Magazine for Raspberry Pi Users!
PACKED with terrific tips, techniques, and projects
raspberry pi GEEK
Photo Tricks!
Time-lapse video and motion detection with the Rasp Pi camera
Create an interactive science display
Live Mad Libs in Scratch
arkOS: Your Pi as a web and mail server
Private Pi
Anonymous surfing with a Rasp Pi Tor Proxy
Coffee Please!
Free DVD!
5 Great Pi Distros
WWW.RASPBERRY-PI-GEEK.COM
New Magazine for Raspberry Pi Users!
PACKED with terrific tips, techniques, and projects
raspberry pi GEEK
BrickPi
Raspberry Pi meets Lego Mindstorms
Talk to your Pi
with a TV remote
Emergency PC
Outfit your Rasp Pi as a backup desktop system
Rasp Pi Hosting
Yeah, right ... Is this for real?
Pi Tricks:
Control a 30-year-old Nikon camera!
Also Inside ...
Free DVD!
We call at Raspberry Towers
Meet Rasp Pi creators
Eben Upton and Gordon Hollingworth
WWW.RASPBERRY-PI-GEEK.COM

Subscribe Today!

Each issue of Raspberry Pi Geek is a new adventure, with ingenious applications and cool projects for Raspberry Pi and the open hardware revolution.



For a limited time only!

Order all three previous issues of Raspberry Pi Geek for only US$ 30*!

* £ 14.99 / € 23.99 / $ 33.50 rest of world

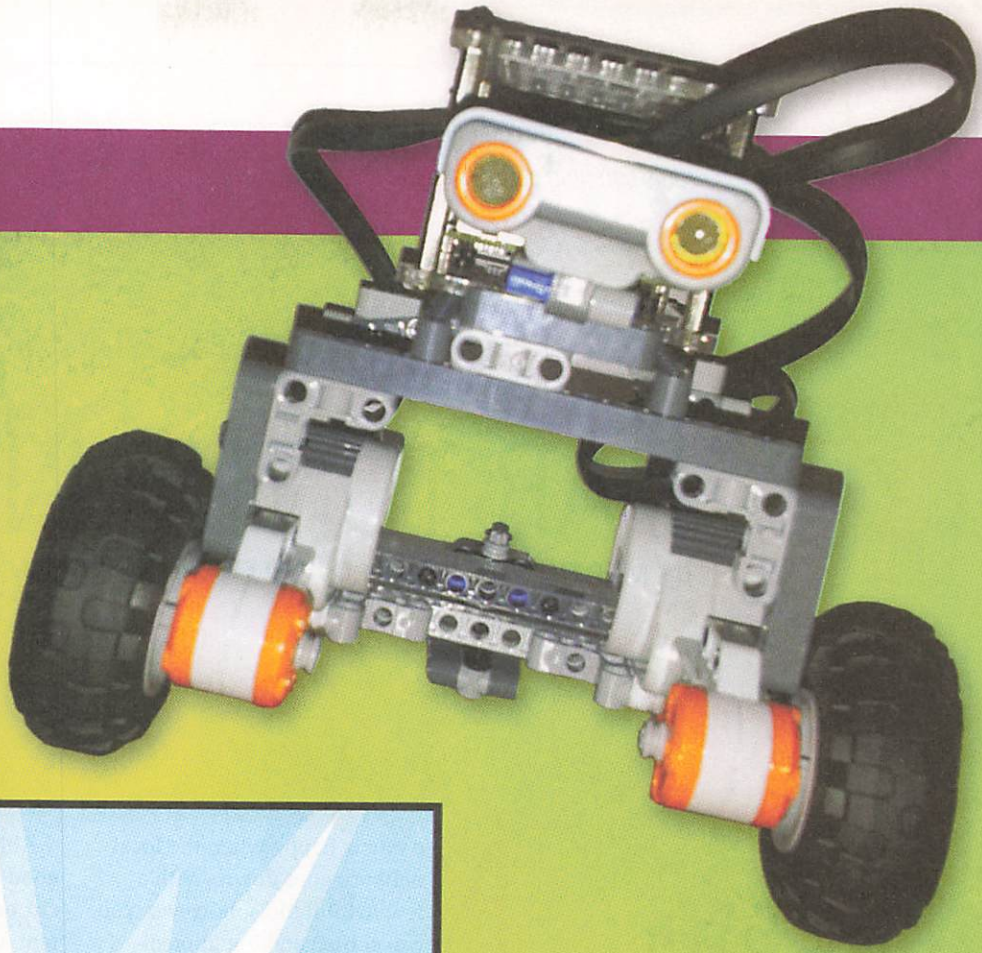


Image © Daniel Villeneuve, 123RF.com

➤ shop.linuxnewmedia.com

Many programs communicate over the network, and when a bottleneck occurs, these tools can help you determine the cause.

Programs for bandwidth monitoring

Ear to the Net

BY MARTIN STEIGERWALD AND FRANK HOFMANN

No one is happy when the local network goes down. With a few helpful tools, however, you can start searching immediately for the nasty LAN-breaking culprit and fix the situation in an instant with a bit of skill and some know-how. Using a combination of iptraf, ifstat, tcpstat, bmon, and pktstat, you can quickly find what's impeding your data traffic through NetHogs [1] and Atop [2].

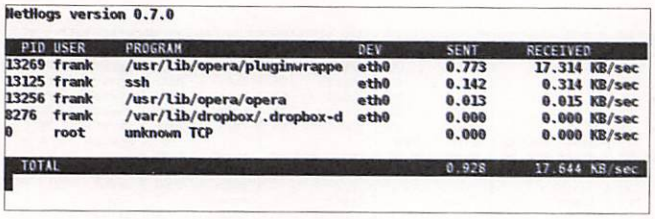
Although real-time analysis tools are good for answering questions about the current state of things on the Linux system, they are less suitable for longer-term evaluation. Answers to questions like "How much traffic did clients and network services produce in the last week" or "Who clogged things up last night at 2 o'clock?" require that you keep the data throughput and statistics constantly in sight.

Programs such as vnStat [3], ntop [4], darkstat [5], and BandwidthD [6] can help you in this endeavor as well as in drilling down into the protocols and processes. Some of the tools also provide a web-based interface and are thus ideally suited for installation on routers or servers, allowing for reliable reporting over time intervals.

On the Fly

If you just want to take a quick glance at the console to see which interfaces are experiencing unusually high traffic, then netstat [7], nload [8] and nicstat [9] are viable options. The first two have been standard tools for Linux for years, whereas the third is a newcomer that comes from (Open)Solaris administrators. Ubuntu packages the first two by default, and the third is available from James Troup's PPA [10].

Nicstat shows the values for each interface (Listing 1, second column), along with received and transmitted data volume per second (third and fourth column), average packets per second (fifth and sixth column) and



```
NetHogs version 0.7.0

  PID USER     PROGRAM                  DEV        SENT      RECEIVED
13269 frank    /usr/lib/opera/pluginwrappe eth0      0.773     17.314 KB/sec
13125 frank    ssh                      eth0       0.142      0.314 KB/sec
13256 frank    /usr/lib/opera/opera     eth0       0.013      0.015 KB/sec
 8276 frank    /var/lib/dropbox/.dropbox-d eth0    0.000      0.000 KB/sec
    0 root     unknown TCP                         0.000      0.000 KB/sec

  TOTAL                                           0.928     17.644 KB/sec
```

Figure 1: NetHogs in use – here with SSH, Opera with audio/video plugin, and the Dropbox network service.

packet size (the last two columns). The last two columns show the degree of use and saturation. Listing 1 clearly shows that most of the data flows over *eth0*, with the largest packets coming over the loopback device *lo*, while the wireless interface *wlan0* seems inactive.

Under the Hood

Running the NetHogs program requires root privileges. At first, it collects only the information for the *eth0* interface, but you can add further options as parameters.

The tool then displays a list of processes based on the network bandwidth (Figure 1). In doing so, it sorts the output in descending order based on usage or load and updates the display every second. With the *-d < seconds >* option you can set another interval, such as *-d 5* for updates every five seconds.

The NetHogs output consists of six columns – process ID (PID), process user, program call, network interface used, and data sent and received over it [11]. Pressing M lets you change the output in the last column from KB/sec to the total amount in bytes, KB, or MB.

Pressing R and S lets you sort the output by received or transmitted data, respectively. You can identify the process running amok by the PID and then end or pause the process if needed.

Currently NetHogs analyzes TCP connections only, not UDP services such as BitTorrent. If the text version doesn't suit you, a BusyTasks [12] plugin is ready to integrate with KDE-Plasmoid that uses NetHogs as a back end.

All in One

In a crunch, there's no avoiding Atop. Along with CPU, memory, and hard drive space use, it indicates the load intensity of each interface as a percentage. It provides a wealth of information about the system as a whole, compared to the rather sparse NetHogs.

Atop requires an additional kernel mode that is not part of the standard kernel's scope. Whereas older versions provided no way of patching and compiling a kernel, version 2 with Netatop [13] allows you to add the feature through a module. Details are in the "Creating and Integrating Atop and Netatop Yourself" box.

After a successful installation, start Atop and press N to change to network view. The *NET* column shows the percentage of network process usage based on the total received and transmitted data.

The columns *BANDWI* (bandwidth incoming) and *BANDWO* (bandwidth outgoing) show the relevant data. The assignment to each process shows up in the first and second columns and include the process ID and program name. The remaining columns show the traffic separately for TCP and UDP packages.

The *TCPSND* and *TCPRCV* fields show the number of sent and received TCP packages; *TCPRASZ* and *TCPSASZ* show their average sizes. The UDP fields follow the same pattern.

In Figure 2, the PlaneShift client, Newsreader aggregator, and an *apt-get update* share the interfaces. The *pscli-ent.bin* PlaneShift client uses small UDP packages, Newsreader gets to the Privoxy private sphere proxy via the HTTP KIO module, and the *http* threads belong to Apt-get.

By default, Atop shows all interfaces bearing traffic and all processes that create it. With S, you can filter by disk, logical drive, and interface; with P, you can filter by process. Atop uses regular expressions for this. Figure 2 shows, for example, that Atop is hiding disks and logical drives.

Time-Based Overview

To get totals of traffic usage or for your own dedicated servers that charge rates for volume of traffic used, you can use programs that display data transfer per day, week, or month, and you can estimate the expected data volume for a time interval.

Atop provides reports for an interval, but it doesn't summarize the data for interfaces. One tool that does is the command-line program vnStat. Technically speaking, vnStat runs with normal user privileges, but it's become customary to write compact binary data to */var/lib/vnstat/< interface >* over the vnstatd service that runs with root privileges.

Use the command *vnstat -u -i < interface >* for each interface to monitor, which accesses a separate database. A *service vnstat start* starts the recording service. Starting the program without parameters, you get an overview like what you see in Listing 2.

The tool logs the data and displays it in columns for each defined interface. Whereas the first column shows the timeframe, the next two show the amount of received and transmitted data, with the fourth column showing the total number. The last column shows the expected data traffic for the given month.

Creating and Integrating Atop and Netatop Yourself

Many distributions don't yet provide the new Atop version, so you need to build Atop and Netatop yourself from sources. Start by downloading the current archive with the source code, along with the module, from the Atop website to a directory. As of this writing *atop-2.0.2.tar.gz* and *netatop-0.3.tar.gz* are the current versions.

Atop currently uses the Zlib and Ncurses libraries, which you can find for Ubuntu-based dsitrbutions in the *zlib1g-dev* and *libncurses5-dev* packages.

After unpacking the Atop source code, compile the program with *cmake* in the unpacked directory. Only the last step – installing the compiled program with *make install* – requires root privileges. Unpack the sources for Netatop in the same way. For compilation, you need the header files for the current kernel from the *linux-headers-<version>* package. You can find the current version number with *uname -r*.

Compile the Netatop source code with *make* and install the module with *make install* as root user. As the final step, load the module with *modprobe netatop*, after which the function is available in Atop. If at some point, you no longer need Netatop, you can uninstall it from the kernel using *modprobe -r netatop*.

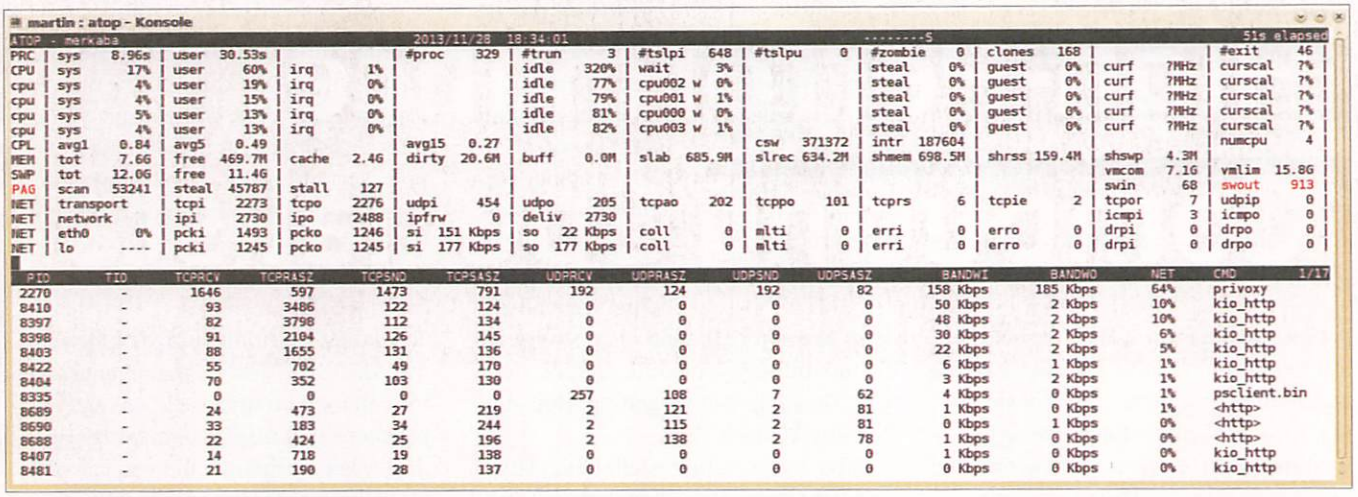


Figure 2: Atop shows how much bandwidth the PlaneShift client, the Newsreader aggregator when updating feeds, and apt-get update need.

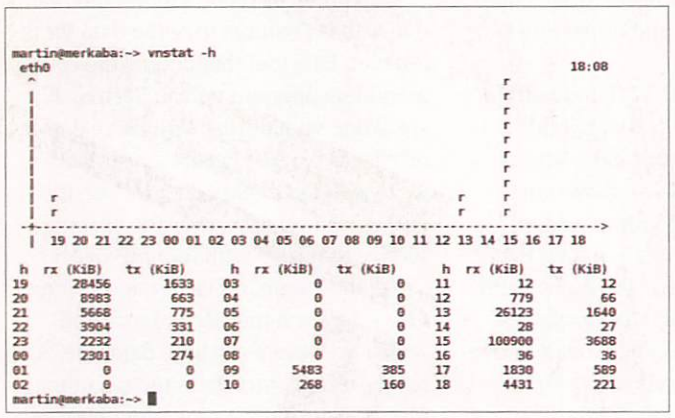


Figure 3: An hourly report from vnStat with a peak data reception between 15:00 and 16:00.

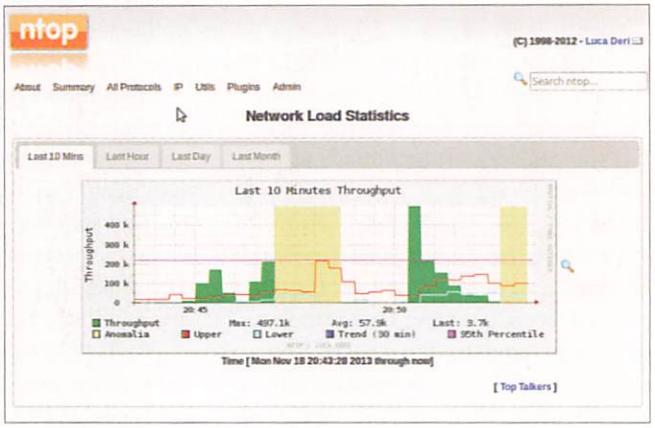

ntop
(C) 1998-2012 - Luca Deri
About Summary All Protocols IP Utils Plugins Admin
Network Load Statistics
Last 10 Mins Last Hour Last Day Last Month
Last 10 Minutes Throughput
Throughput
Throughput Anomalia Upper Lower Trend (30 min) 95th Percentile
Time [Mon Nov 18 20:43:28 2013 through now]
[Top Talkers]

Figure 4: Ntop provides data throughput statistics as a chart.

With regard to mobile access, you're advised to check with your provider regularly so that differences in counting methods are caught in time. A little buffer up to the maximum limit doesn't hurt to stay on the safe side.

By default, vnStat shows the data for the past two days and a summary of the month. Use the *-i < interface >* option to restrict the output to the given interface. The *-w* option provides a weekly overview; with *-h* you get an hourly report that includes an ASCII graph (Figure 3). An interface at more than 90 percent capacity is highlighted in red.

Listing 1: Nicstat Displays Data Volume, Packet Size, etc.

```
merkaba:~> nicstat
    Time      Int   rKB/s   wKB/s   rPk/s   wPk/s    rAvs    wAvs %Util    Sat
19:53:07     eth0    2.39    0.11   11.64    0.88   210.4   127.4  0.02   0.00
19:53:07       lo    0.34    0.34    0.20    0.20  1719.8  1719.8  0.00   0.00
19:53:07    wlan0    0.01    0.00    0.05    0.00   169.7   155.4  0.00   0.00
```

Listing 2: Overview with vnStat

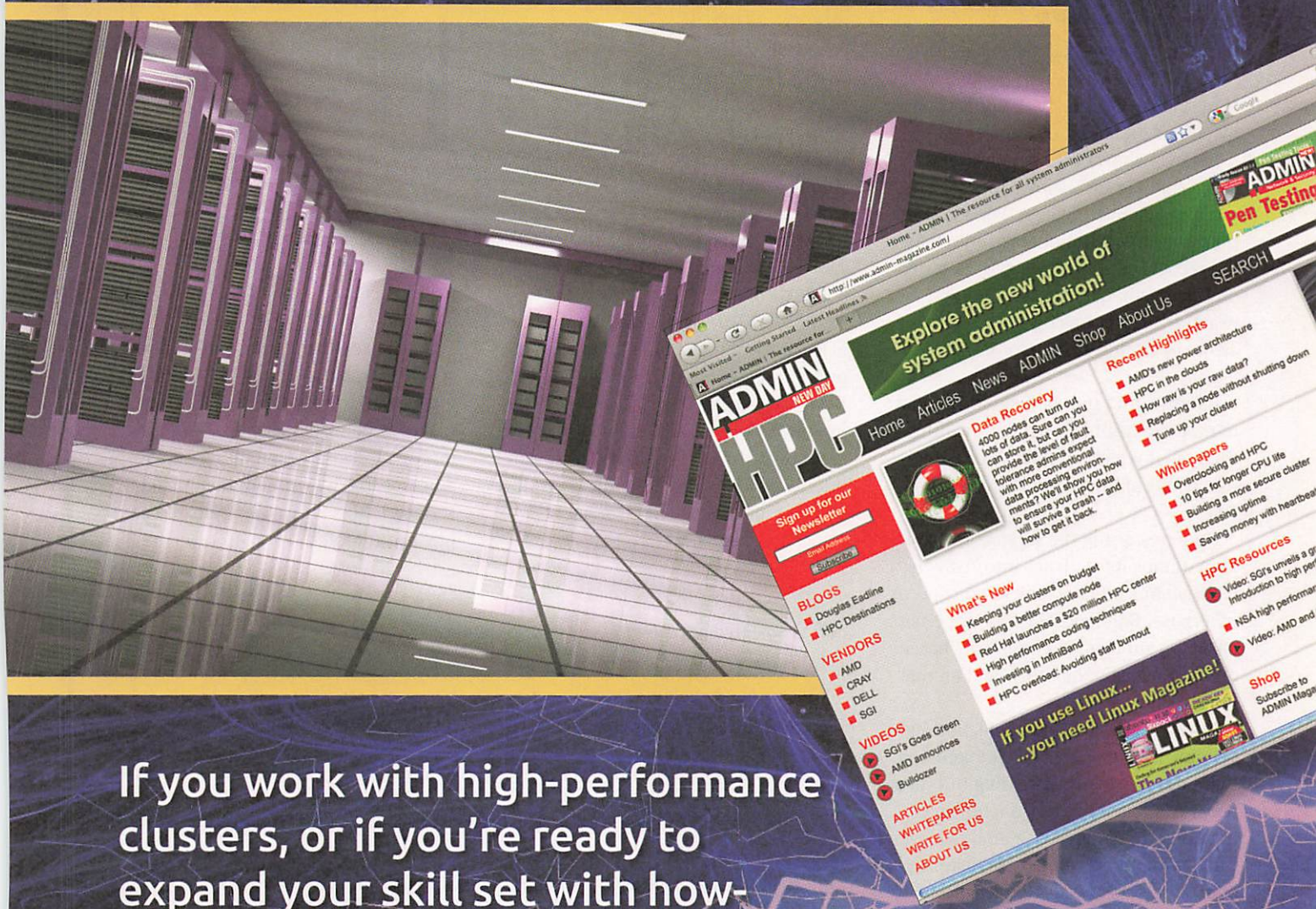
```
# vnstat

                 rx      /     tx      /    total    /  estimated

 eth0:

     Dec '13    3.75 GiB  /  298.87 MiB  /   4.04 GiB  /    7.67 GiB
   yesterday  110.05 MiB  /   12.09 MiB  /  122.14 MiB
       today  134.50 MiB  /    6.67 MiB  /  141.17 MiB  /     188 MiB


 wlan0:

     Dec '13   11.35 MiB  /    1.42 MiB  /   12.77 MiB  /   21.00 MiB
   yesterday       0 KiB  /       0 KiB  /       0 KiB
       today       0 KiB  /       0 KiB  /       0 KiB  /      --
```

Web-Based Programs

A good selection of web-based programs is available for monitoring complex server environments (e.g., Munin [14], Cacti [15], Zabbix [16], Zenoss [17], Bloonix [18], Nagios [19], and Splunk [20]). However, for single-user systems and special cases, ntop, BandwidthD, and darkstat are viable alternatives.

The ntop service collects data on system activities as a background task. Port 3000 provides access to an interface over which you call up relevant data.

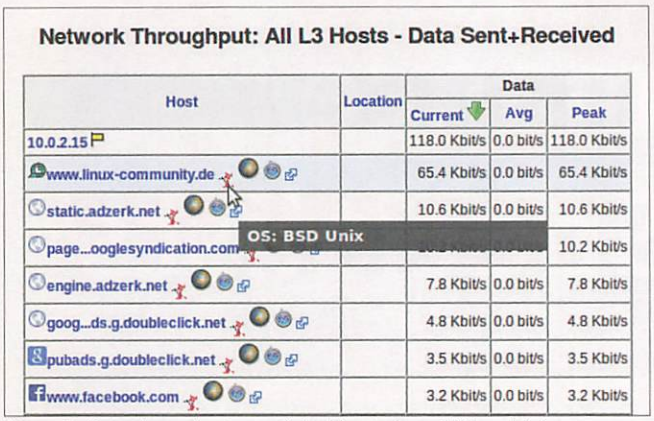
A Webzine for High-Performance Computing Specialists



If you work with high-performance clusters, or if you're ready to expand your skill set with how-to articles, news, and technical reports on HPC technology,

http://hpc.admin-magazine.com

ADMIN Network & Security


Network Throughput: All L3 Hosts - Data Sent+Received
Host
Location
Data
Current
Avg
Peak
10.0.2.15 118.0 Kbit/s 0.0 bit/s 118.0 Kbit/s
www.linux-community.de 65.4 Kbit/s 0.0 bit/s 65.4 Kbit/s
static.adzerk.net 10.6 Kbit/s 0.0 bit/s 10.6 Kbit/s
page...ooglesyndication.com 10.2 Kbit/s
OS: BSD Unix
engine.adzerk.net 7.8 Kbit/s 0.0 bit/s 7.8 Kbit/s
goog...ds.g.doubleclick.net 4.8 Kbit/s 0.0 bit/s 4.8 Kbit/s
pubads.g.doubleclick.net 3.5 Kbit/s 0.0 bit/s 3.5 Kbit/s
www.facebook.com 3.2 Kbit/s 0.0 bit/s 3.2 Kbit/s
Figure 5: Data throughput per visited host plus additional info.

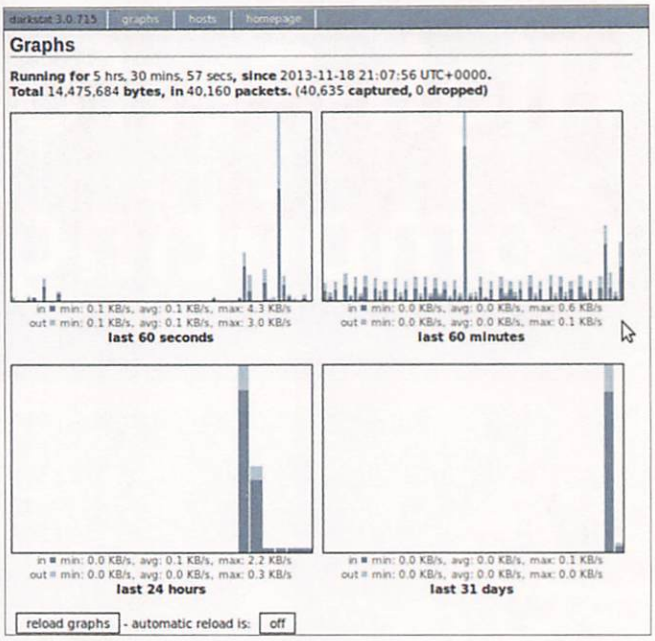


Figure 6: Darkstat shows a bar chart for incoming and outgoing data traffic.

Ntop filters the data by IP address, protocol, host/domain, or transfer rate.

Figure 4 shows the statistics as a chart, and Figure 5 shows the throughput for each visiting host. For each computer, you see additional data such as the recognized operating system based on fingerprint detection. In Figure 5, for example, the tool recognized *BSD-Unix* as the operating system for *linux-community.de*.

BandwidthD and darkstat are similar to ntop. To begin, install BandwidthD then tell it what you'd like to observe – that is, the subnets and network interfaces.

Upon restarting the service, the daemon dumps its statistics into */var/lib/bandwitdhd/htdocs/*, unless you specify otherwise. BandwidthD provides charts based on subnets and interfaces based on daily, weekly, monthly, and annual intervals.

Darkstat is small, portable, and lightweight. It listens on port 666 and analyzes data not only over time but also per hosts receiving or transmitting data. Figure 6 shows the analysis over time, which you can get consistently over the web browser. At this point, darkstat doesn't understand IPv6, however.

The program starts only after you have successfully configured */etc/darkstat/init.cfg*. There you specify the listening interface, the address range of the local network, and the IP address of the listening web interface. Follow up with *START_DARKSTAT = "yes"*.

Safety Measures

With web interfaces, you will want to consider how much to allow outside access. Ntop provides the settings for Debian-based distributions, such as Ubuntu, in the */etc/default/ntop* file. Note that it uses the loopback address *127.0.0.1* on port 3000 (*GETOPT = "-w 127.0.0.1:3000*) by default.

For Darkstat set the IP address and port via */etc/darkstat/init.cfg*. The value *BINDIP = "-b 127.0.0.1"* specifies listening locally only. The BandwidthD program does not come with its own web interface. To access a graphical display, you can install a web server such as Apache or Nginx [21].

Conclusion

You can use the tools described in this article to track programs that are affecting performance, so that no troublemaker remains undetected. Because there's no Swiss Army knife that covers all cases, the web-based programs described in this article can help you complete your toolkit. •

Info

[1] NetHogs: *http://nethogs.sourceforge.net/*
[2] Atop: *http://atoptool.nl*
[3] Vnstat: *http://humdi.net/vnstat/*
[4] Ntop: *http://www.ntop.org*
[5] Darkstat: *http://unix4lyfe.org/darkstat/*
[6] BandwidthD: *http://bandwidthd.sourceforge.net*
[7] Netstat: *http://en.wikipedia.org/wiki/Netstat*
[8] Nload: *http://www.roland-riegel.de/nload*
[9] Nicstat: *http://sourceforge.net/projects/nicstat/*
[10] Nicstat PPA: *https://launchpad.net/~elmo/+archive/nicstat*
[11] NetHogs: *http://manpages.ubuntu.com/manpages/jaunty/man8/nethogs.8.html*
[12] BusyTasks: *http://kde-apps.org/content/show.php?content=143833*
[13] Netatop: *http://www.atoptool.nl/netatop.php*
[14] Munin: *http://munin-monitoring.org*
[15] Cacti: *http://www.cacti.net*
[16] Zabbix: *http://www.zabbix.com*
[17] Zenoss: *http://www.zenoss.com*
[18] Bloonix: *http://www.bloonix.net* (in German)
[19] Nagios: *http://www.nagios.org*
[20] Splunk: *http://www.splunk.com*
[21] Nginx: *http://www.nginx.org*

DON'T MISS A SINGLE ISSUE!

Ubuntu User is the first print magazine created specifically for Ubuntu users. Ease into Ubuntu with the helpful Discovery Guide included in each issue, or advance your skills with in-depth technical articles, HOW-TOs, reviews, tutorials, and community reports.

SUBSCRIBE NOW!
4 issues per year for only
£ 24.90 / EUR 29.90 / US$ 39.95

- ✔ Don't miss a single issue!
- ✔ Huge savings – Save more than 35% off the cover price!
- ✔ Free DVD – Each issue includes a Free DVD!

DVD ROM
Kubuntu 13.10 + Linux Mint 16
Two fantastic Ubuntu-compatible distros on one double-sided dvd!
Kubuntu + Mint
THE 2 BEST UBUNTU DERIVATIVES
UBUNTU user
EXPLORING THE WORLD OF UBUNTU
ROAMING THE WORLD WITH UBUNTU
EXPLORERS
ANDROID APPS
Create and test your own with Genymotion
Geotag your photos
Explore the world from your desktop
Manage your trips with MyTourbook
Build your own off-road

shop.linuxnewmedia.com

Some Linux followers see systemd as the best thing since sliced bread – for others, it's the work of the devil. However, it also has the stuff to clear out old trenches and form a unified base for Linux.

Systemd as central control for the Linux system

Strong Glue

BY FERDINAND THOMMES

Systemd is the result of the work of many developers, but, if there's a public face to be given to the software, it's certainly that of Lennart Poettering.

The valiant developer has already checked off a number of important projects on his to-do list, including Avahi and the PulseAudio sound system. With systemd, he's been working since 2010 on getting central services on a modern, uniform basis – emanating from a new approach for the init system.

Poettering doesn't shy away from unpopular decisions, which has earned him the reputation of using dictatorial methods. The decision to develop systemd only for Linux brought criticism, scorn, and insults. For him, meanwhile, the decision has a clearly understandable foundation.

Along with his work on systemd, Poettering is advocating for a unified Linux desktop. His newest project, which he's pursuing together with kernel guru Greg Kroah-Hartman, deals with integrating the Dbus message system as Kdbus in the kernel. His announcement at Google Plus [1] that systemd, together with Kdbus and all userspace tools in Fedora, starts without a problem and, thus, represents a new milestone, resulted in yet another hailstorm of harsh criticism.

Such opposition, however, has never diverted the Red Hat developer from his course (see the "Lennart Poettering Interview" box). He likes to speak his mind (Figure 1) as a free-thinker with enough backbone and support to see his vision through and perhaps remove a few old Linux relics.

Cadaverhan, sxc.hu

To the Roots

If you're dealing with an init system, you very quickly recognize why working on this key component can cause so much turmoil. The term comes from "initiating" and encompasses, on Unix-like operating systems, the system startup and orderly shutdown.

Init is the first process the kernel generates, and it starts all other child processes. It's therefore assigned process ID 1 (PID 1). Among init's tasks are essentially starting all necessary services, mounting the filesystem, and setting up the network.

The SysVinit [2] init system used on Linux and other Unix derivatives since the

1990s was developed in 1988, when AT&T needed a start mechanism for its System V operating system in version R4.

In 2004, Solaris replaced SysVinit with its successor, Service Management Facility (SMF) [3], followed for the Mac OS X in 2005 by launchd [4]. Canonical has used Upstart [5], developed in-house based on SysVinit, since 2007. With Gentoo's FreeBSD offshoot, OpenRC [6] has been an alternative since 2007.

Born of Necessity

Systemd essentially comes from Red Hat developers Lennart Poettering and Kay Sievers. Before the two set upon the task of remaking the init system under Linux in 2007, they took a close look at launchd and SMF, because these init systems provided parallel processing startup capabilities that the static SysVinit didn't have.

Because Upstart also had this ability, the original idea of the developers was to enhance Ubuntu's init system. Legal reasons that had to do with Canonical's Contributor License Agreement (CLA) [7] prohibited this – in that neither Poettering nor Sievers was willing to sign the agreement.

Thus, systemd was born of necessity. Two fundamental decisions of the time still raise hackles: Systemd was deliberately and without compromise designed for Linux only. Systems with other kernels such as BSD and Hurd simply stood on the sidelines; a porting to BSD was possible, but expensive.

The other decision grated against the old Unix principle of "one tool for one job"

Figure 1: Lennart Poettering's t-shirt shows his opinion on the sayings of Mark Shuttleworth.

Lennart Poettering Interview

Linux User: *Why do you think systemd is viewed so critically, even though it's implemented more consistently than some other software we use every day?*

Lennart Poettering: Systemd may evoke a controversy, but overall the assessment has been far more positive than negative. The recent decision from Debian in favor of systemd makes it clear that not only do individual distributions opt for the approach, but the clear majority of all major distributions do as well.

Of course, there is a significant minority in the community that criticizes or even totally rejects systemd. The reasons for it vary. A big part of it is that the concept cuts off some old heads and represents a major change in the classic Unix system.

Another reason lies in the fact that systemd not only optimizes the actual boot process, but unifies many of the system's low-level components.

Still others in the community reject it for economic reasons. Thus, Canonical has put a lot of effort into developing Upstart and was understandably interested in seeing to its own project's success.

LU: *Debian, after much strife, has settled on some sensitivities for systemd. Do you see the last few months as a democratic process or simply as evidence of a lack of a benevolent dictator?*

LP: The various distributions are organized differently. Debian is set on a democratic process, Ubuntu rather has the concept of a "benevolent dictator." Fedora is somewhere in the middle, a kind of technocracy.

Certainly the debate on this subject was largely unpleasant at Debian. But after all the chaotic discussion, it remains to be determined that this exactly constitutes the democratic process and that democracy works perfectly well.

The process takes its time and is sometimes no fun, but this is the deal: If we want to create a universal operating system, it means that we have to bring in a lot of people, which necessarily leads to extensive discussion and a wider range of contributions – positive as well as negative.

LU: *Systemd is to extend further into userland, and you want to start processes in the desktop environment with the same mechanisms as the system itself. Will systemd be brewing our coffee?*

LP: What exact tasks belong to systemd and which don't is debatable. There are few modules we're still integrating. Generally speaking, components of what we call the Plumbing Layer come into play – not the kernel or interface, but the "glue" between them.

For us, it's always the general usability, so we want a generic solution, not a specific one. Systemd is thus never a product, but something from which other developers can build products. At best, it should hold the system together without users being in direct contact with it.

To come back to the coffee. According to this definition, systemd is certainly a useful component, among many others, in building an Internet-capable coffee machine – but it never brews the coffee. That would be too specialized and no longer the "glue" that holds the coffee machine together.

I personally favor Club-Mate as the coffee source. That means that there's nothing further from my mind than to build coffee machine functionality into systemd.

LU: *Not only your software, but you yourself are controversial. You deal with it pretty casually. Do you understand why so many criticize you personally, and how does that affect you?*

LP: Anyone trying to swap out a key component of the Linux ecosystem is open for criticism; that's unavoidable. The community does have strong convictions and if you contradict them, it's easy to become a bogeyman.

Systemd was not the first project that tried to turn the boot system inside out – there've been dozens. We've often asked ourselves why we've been successful in advocating our approach. We'd like to believe it's based on technical grounds. At least in our view systemd definitely represents the better system.

How we've managed to deal with the community's pressure also plays a major role. My colleagues and I have probably been better at maintaining a good sense of humor and developing clear convictions, and that with a deep breath.

The Linux community sometimes behaves badly. If you want to set something in motion, you can't tread too softly. It would certainly be nice if it weren't that way!

in that it set some of the jobs pretty close together. This may loosen up slightly in the future so that functions are available only individually, as Ubuntu currently does with Upstart in conjunction with logind.

More than Init

Systemd is short for system daemon, that is, a background process that provides services for the system. The init process was at the beginning of the development and still gets most of the attention. Other components such as logind and the journal, however, are up there as well.

Systemd's init process is based on a slim PID-1 service that does only the bare minimum. Thanks to parallel processing, the computer's startup and shutdown runs much faster. It's based on a technology that Apple developed for launchd: socket activation.

Socket activation allows all services to start concurrently. The process also got simpler because developers and administrators no longer need to be aware of which service starts at which point during bootup with which dependencies. The trick is that systemd itself creates the communication sockets with the services to be started and buffers the data until the service is running and accepts it. The SysVinit concept rather forces dependent services to wait for each other, because the init system doesn't store any data itself.

Because all services start concurrently, systemd contributes greatly to simplifying the structure of the service files. These replace the SysVinit shell scripts that often went wild. Figures 2 and 3 show the differences based on the Kexec [8] example.

Systemd organizes all tasks pending at startup – such as readying the hardware, mounting the filesystem and starting the services – in units. The service files of the various units have their own filename extensions, which give a clue as to their functions.

Files with the *.mount* extension take care of mounting and unmounting the filesystems; those with the *.service* extension handle the background services. These units are stored uniformly in */lib/systemd/system*. For a modification, copy the files in question manually to */etc/systemd/system/* and edit them as needed; the system automatically uses them from then on.

An important decision in designing systemd that contributed to its being used only in Linux was the integration of cgroups [9]. These control groups allow allocating and limiting kernel resources, such as CPU time, data throughput, and process group memory usage.

Systemd therefore provided new unit types as of version 205, including scope units and slice units. The former collect the system services or applications and all their dependencies into cgroups and manage them. The slice units provide the units with the required resources.

In the long term, the developers want to hand over the use and control of the cgroups completely to the daemon. In this way, systemd would become a new intermediate layer between the kernel and applications. Poettering himself sees it as a "glue." In any case, the structure thereby acquires some complexity (Figure 4).

Listing 1: Results of systemd-analyze

```
01 Â§Â§nonumber
02 Startup finished in 10.432s (kernel) +
   6.287s (userspace) = 16.720s
03 Startup finished in 1.327s (kernel) +
   3.116s (userspace) = 4.443s
```

Journal

Unlike SysVinit, systemd logs the boot messages right from the beginning. Thus, screenshots of the boot process captured by smartphone for support forums become a thing of the past. Systemd instead creates a socket [10] for it at runtime, on which a minimal logging service runs and records the messages from the early boot phase. It later passes the socket to the regular protocol services and the message is journaled.

The journal replaces the long-used Syslog. In a first demonstration, it received much criticism because the journal deviated from the ongoing concept of storing text files in a flat hierarchy. Instead the files are stored as binary.

Apart from that, the journal has some hard-to-ignore advantages. Besides the aforementioned boot process logging, it allows for highly targeted queries. Thus, the command on the first line below shows the messages since the last system startup; whereas the second line shows what the web server put out in messages since midnight. In a blog entry, Poettering shows other more detailed queries [11].

```
$ journalctl -b
$ journalctl -u httpd
```

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          kexec
# Required-Start:
# Required-Stop:     reboot
# X-Stop-After:      umountroot
# Default-Start:
# Default-Stop:      6
# Short-Description: Execute the kexec -e command to reboot system
# Description:
### END INIT INFO

PATH=/sbin:/bin

. /lib/lsb/init-functions

test -r /etc/default/kexec && . /etc/default/kexec

do_stop () {
    test "x`cat /sys/kernel/kexec_loaded`" = "x1" || exit 0
    test -x /sbin/kexec || exit 0

    log_action_msg "Will now restart with kexec"
    # Clear the screen if possible
    printf "\033[H\033[2J"
    /sbin/kexec -e
    log_failure_msg "kexec failed"
}

case "$1" in
  start)
    # No-op
    ;;
  restart|reload|force-reload)
    echo "Error: argument '$1' not supported" >&2
    exit 3
    ;;
  stop)
    do_stop
    ;;
  *)
    echo "Usage: $0 start|stop" >&2
    exit 3
    ;;
esac
exit 0
```

Figure 2: SysVinit: The Kexec startup script elicits constructs at various places where the purpose isn't readily apparent.

```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as published by
#  the Free Software Foundation; either version 2.1 of the License, or
#  (at your option) any later version.

[Unit]
Description=Reboot via kexec
Documentation=man:systemd-halt.service(8)
DefaultDependencies=no
Requires=shutdown.target umount.target final.target
After=shutdown.target umount.target final.target

[Service]
Type=oneshot
ExecStart=/bin/systemctl --force kexec
```

Figure 3: Systemd: The Kexec service file needs only a few lines whose function is clearer based on the variable names.

```
--since=00:00 --until=9:30
```

As the name logind [12] suggests, the daemon takes care of user and session management and replaces the no longer developed ConsoleKit [13]. Thus, systemd begins to enter userspace, which again was not met without criticism.

In the end, Poettering wants to go still further by using the same mechanism for the processes starting Gnome or KDE and its applications. Apple's Mac OS X already does this. The first approaches can already be seen in Gnome 3, which has recently been pulling in parts of systemd by integrating logind.

Neverending Story

Distributions such as Fedora and openSUSE, but also Debian derivatives such as Siduction, Semplice, and soon Tanglu, have already switched over to systemd (Figure 5).

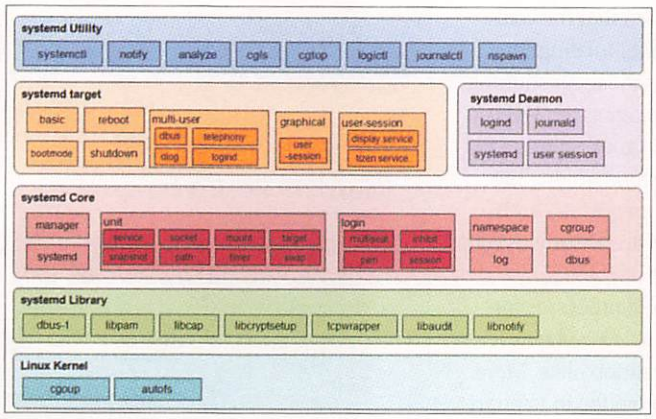


Figure 4: The different layers of the systemd infrastructure make it difficult to consider it a simple service.

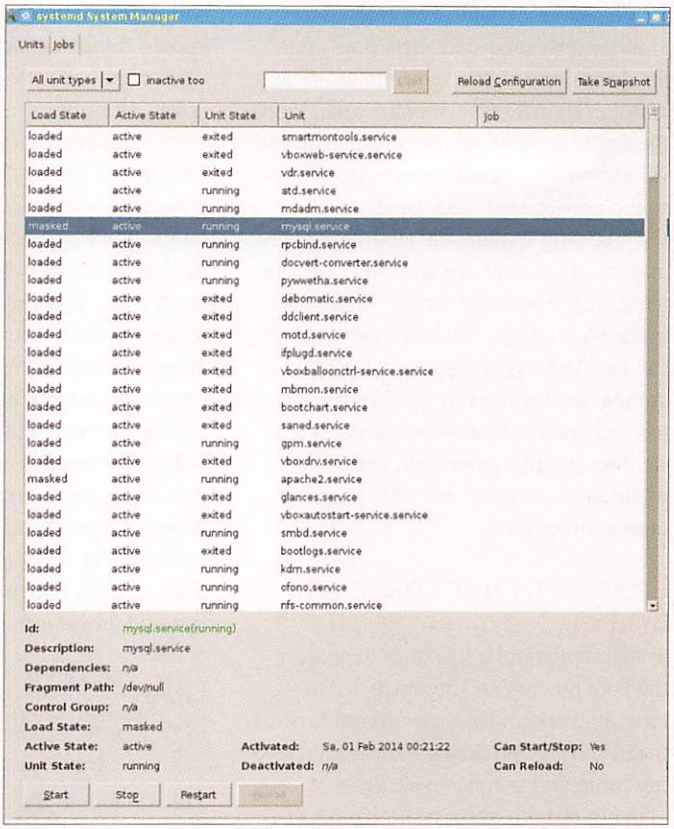


Figure 5: Front ends already exist to exhibit systemd activities.

The dependencies to parts of systemd were the catalyst for hard decisions about the future init system in Debian. The discussion began in October 2003 with news on the Debian developer mailing list [14]: A complaint was filed about the dependency of the Gnome settings daemon on systemd – introduced by the hope that it wouldn't "turn into some kind of flame war."

But, that's just what happened. The discussion became heated and with no result, until it became clear after a week that no solution was possible. Another developer requested that the Debian Technical Committee (CTTE) [15] should take on the problem [16] and come up with a decision [17].

The CTTE familiarized itself with the technical issues and tried to find a solution supported by all its members. The board comprised eight members – of which four soon supported systemd, with the other four supporting Upstart. Of the latter, two were currently working at Canonical, and a third was a former Ubuntu employee.

The Debian project has its particular problems because of its many supported architectures. Thus, it was clear from the beginning that the Hurd and KFreeBSD platforms weren't working with systemd. This proved to be a problem technically and politically for the largest distribution that decides things without any corporate backing.

Of the three candidates that could replace SysVinit, only one could come to the rescue: OpenRC, developed in Gentoo. Although it currently has little chance of establishing itself as the default init system, with some effort, it could run both architectures.

In the case of Upstart, a fork of the software would apply, because Canonical only allows collaboration when the aforementioned CLA is signed (see the "Canonical's Gag" box). Mark Shuttleworth would probably rub his hands if Debian took over the further development of Upstart.

Early February saw a second set of ballot initiatives [21] that committee member Ian Jackson (a Canonical colleague) proposed, without resolution, only to be followed by a third. The third vote was like the first, a simple proposal – with one difference: It included a clause that allowed Debian developers to adopt the ultimate decision-making tool at Debian, the General Resolution (GR) and thereby repeal the decision of the CTTE with a simple majority [22].

Normally, such a decision needs a two-thirds majority at Debian. The GR needs six official Debian developers to agree. In mid-February, the third ballot led to almost four months of discussion. Finally, Bdale Garbee, chair of the committee, broke the stalemate by invoking for the first time a special voting procedure for such situations and declared systemd a winner [23].

Whether the developers tilt the vote again with another GR has yet to be seen. Systemd would probably win again but would have a broader base. Thus, the argument of an arbitrary decision would be removed from the hands of any critics, with Upstart having hardly a chance in the process. The procedure, however, showed that the CTTE statutes might require some reform. Attempts by one or the other members to drag out the process and torpedo it are hard to overlook.

Canonical's Gag

The controversial Canonical CLA allows having software under a dual license. The developer might retain his copyright, but gives Canonical the right to put the code under another license and no longer develop it further under the original one.

In the case of systemd, the discussion over Canonical in general and CLAs in particular was rekindled. Not only did Sievers and Poettering express themselves, but so did Linux Torvalds [18], Greg Kroah-Hartman [19], and Matthew Garrett [20] in some detail.

Whereas Garrett didn't find CLAs to be all that bad – after all, the Apache Foundation uses one of them – Torvalds contradicted him and considered these agreements as "fundamentally broken." Systemd co-developer Sievers brought his pragmatic view of things sarcastically to the fore: "Without CLA we would still be working on Upstart instead of getting sensible things done on Systemd."

In his blog, Canonical founder Mark Shuttleworth conceded a few days after the decision to be the "graceful loser" and announced that even Ubuntu would make the switch to systemd in the long run, provided developers declared it to be as stable as the previous solution, Upstart.

Thus, with Debian and Ubuntu, the last major distributions are on course with systemd. Together they form a critical mass to influence its development in keeping with Debian. There may also be less contention among the distribution warehouses. Gentoo and Slackware are the sole outsiders; Gentoo continues to rely on OpenRC, while Slackware will probably keep the faith with SysVinit.

Optimizing the Boot Process with Systemd

Systemd provides a few tools to analyze the boot process and optimize it. These tools all work under a user account, which means no root privileges – the corresponding command is *systemd-analyze*.

In our test, on a workstation with KDE and a good many services and SSD for the whole system, the result was not convincing (Listing 1, line 1). A freshly installed current notebook with KDE started faster (Listing 1, line 2). On a lighter desktop environment, the values were in the three-second range.

If the startup takes too long, you can use *systemd-analyze blame* to test which processes and services are to blame. On an older notebook, the *blame* option revealed that the NTP service was stuck in a loop for more than 30 seconds. As it turned out, it was trying to contact a server that no longer existed, so a configuration correction fixed the problem.

If necessary, you can create a graph with *systemd-bootchart* that shows the behavior at boot time (Figure 6). The command is as follows:

```
$ systemd-analyze plot > test.svg; ↵
  inkscape test.svg
```

In this case, the Inkscape application helped in viewing the graph. Alternatives include image viewers such as Gthumb or the Firefox web browser.

Conclusion

The switchover to systemd does not present high hurdles to either distributions or users. In the near future, systemd remains compatible with Sysvinit so that the well-known commands for manipulating services work just as well as those for shutting down, restarting, and changing run levels.

Systemd proves to be a good solution for Debian and Ubuntu, to stand among the others, such as Fedora, openSUSE, Arch Linux, and Mageia, who are already using it. If the developers have any voice, systemd will help standardize processes for all distributions, providing advantages for developers and administration alike. More details on systemd are provided in Lennart Poettering's blog [24].

Some developers are accusing Red Hat of striving for more control over Linux with systemd. Upstart, on the other hand, would pull in a system that is totally under a company's control. Viewed objectively, systemd provides the best technically suited init system for the foreseeable future. Debian would be best to recognize this and be up front about it.

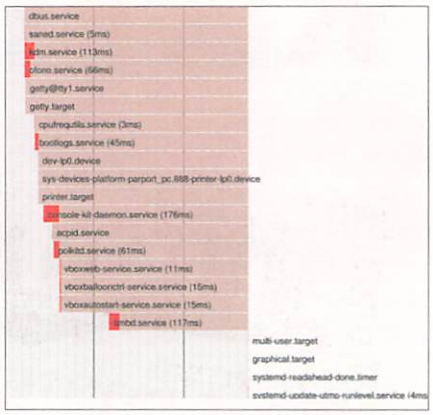


Figure 6: Using a boot chart, you can tell at first glance which services are coming up on startup.

Info

[1] Poettering on Kdbus: *https://plus.google.com/u/0/+LennartPoetteringTheOneAndOnly/posts/13JZ7GpyVDb*

[2] SysVinit: *http://en.wikipedia.org/wiki/Init#SysV-style*

[3] SMF: *http://en.wikipedia.org/wiki/Service_Management_Facility*

[4] Launchd: *http://de.wikipedia.org/wiki/Launchd*

[5] Upstart: *http://de.wikipedia.org/wiki/Upstart*

[6] OpenRC: *http://en.wikipedia.org/wiki/OpenRC*

[7] CLA: *http://en.wikipedia.org/wiki/Contributor_License_Agreement*

[8] Kexec: *http://en.wikipedia.org/wiki/Kexec*

[9] Cgroups: *http://en.wikipedia.org/wiki/Cgroups*

[10] Socket: *http://en.wikipedia.org/wiki/Network_socket*

[11] Journal: *http://0pointer.de/blog/projects/journalctl.html*

[12] Logind: *http://www.freedesktop.org/software/systemd/man/systemd-logind.service.html*

[13] ConsoleKit: *http://www.freedesktop.org/wiki/Software/ConsoleKit/*

[14] Developers list discussion: *https://lists.debian.org/debian-devel/2013/10/msg00444.html*

[15] Debian Technical Commmittee: *https://www.debian.org/devel/tech-ctte*

[16] CTTE request: *https://lists.debian.org/debian-devel/2013/10/msg00703.html*

[17] Bug report to CTTE: *https://lists.debian.org/debian-devel/2013/10/msg00825.html*

[18] Torvalds on CLAs: *http://www.muktware.com/2014/01/linus-torvalds-cla-fundamentally-broken/19811/2*

[19] Kroah-Hartman on CLAs: *https://plus.google.com/111049168280159033135/posts/NstZfwXbAti*

[20] Garrett on CLAs: *http://mjg59.dreamwidth.org/29160.html*

[21] Second set CTTE: *http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=727708#5389*

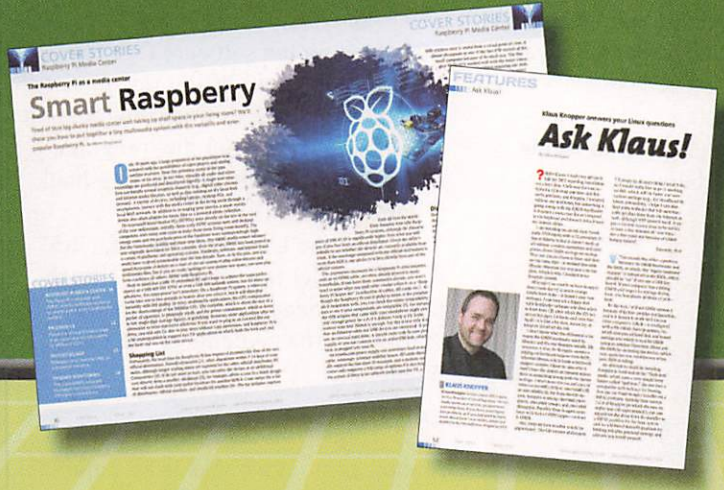
[22] General Resolution: *http://www.debian.org/vote/*

[23] Final decision of CTTE: *http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=727708#6734*

[24] Blog on systemd: *http://0pointer.de/blog/projects/systemd.html*

RISK-FREE TRIAL!

Practical.
Technical.
Elegant.



SUBSCRIBE NOW

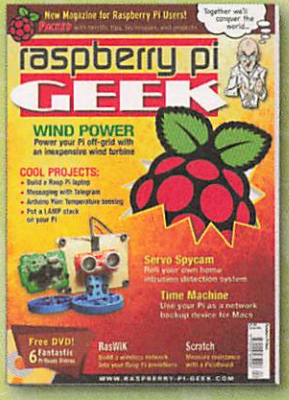
3 ISSUES
+ 3 DVDs

$3

shop.linuxnewmedia.com

£3 / €3 / $3 / $9 rest of world
Depending on the region you live in.

OTHER TITLES AT
shop.linuxnewmedia.com



Calibre lets you bring some order to your digital library. You not only can restore reading material lost on your hard drive, but you can also convert digital books into a format of your choice.

Organizing and reading e-books with Calibre

Bookcase

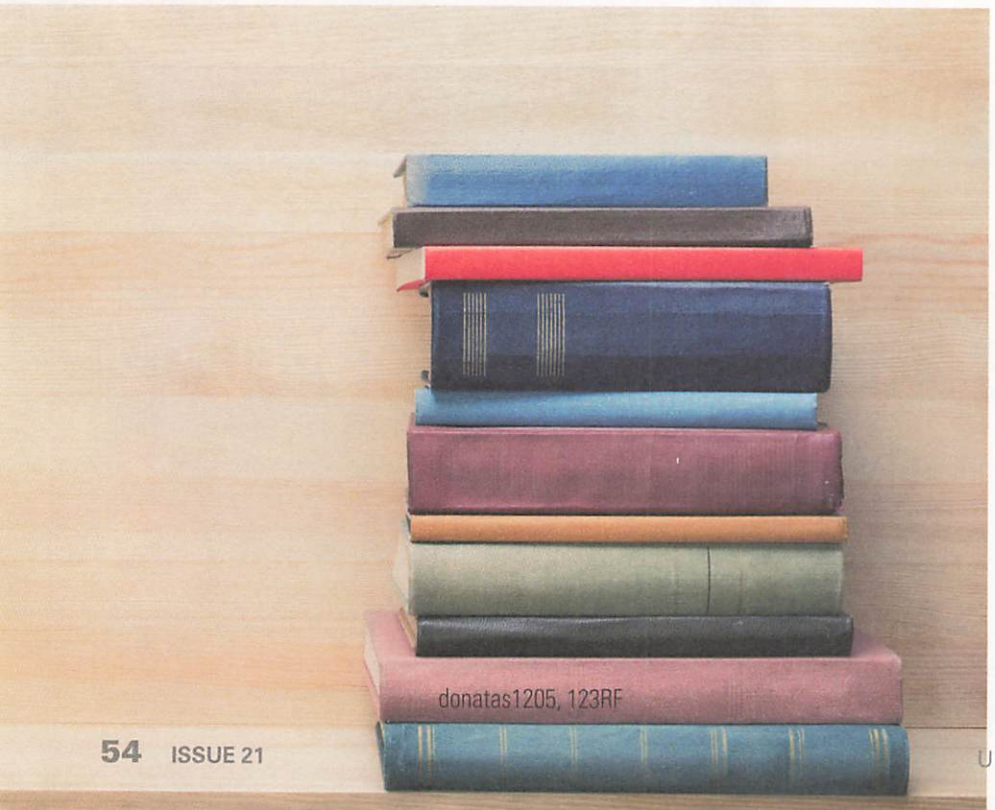
BY DR. KARL SARNOW

If you read a lot of books, you know the problem: Where do they go? Did you classify your copy of *Just for Fun* in biographies, computer science, or science fiction?

Do you really classify books? Who are you anyway? Sheldon Cooper? And, if the book's not in your bookcase, did you lend it to someone or is it propping up a table somewhere? If you're not a librarian with OCD, you'll probably have to ransack you whole house to come up with that elusive paperback title you so desperately want to reread.

It's pretty much the same with e-books. They're guaranteed to reside within the four metal walls of your computer, but trying to find them on the hard drive, because of the pure volume and disorganization of your collection, will not be how you want to spend your Sunday afternoon.

This is where Calibre comes in. The program manages e-books in various folders and displays them for reading on your request. In this article, I'll took a closer look at the newest version 1.11 on Ubuntu 12.04 LTS.



Chapter One

After loading the software (see the "Installation" box), call it up either on the command line or from the program menu. On first startup, the program manual is available only as an e-book (Figure 1).

To open the book and read it, click *Formats* under the image of the book title. This opens a reading window that shows the contents of the book (Figure 2).

Your Own Books

Adding your own books is an important function in helping you navigate through your library. Click the book icon on the left to get a selection dialog.

The selection is impressive in regard to supported formats. Apart from EPUB, LRF, HTML, LIT, MOBI, Topaz, Text, PDF, SNB, and digital comics, Calibre recognizes various text processing documents that it reads in from diverse archive versions. Once you successfully add a book, it ap-

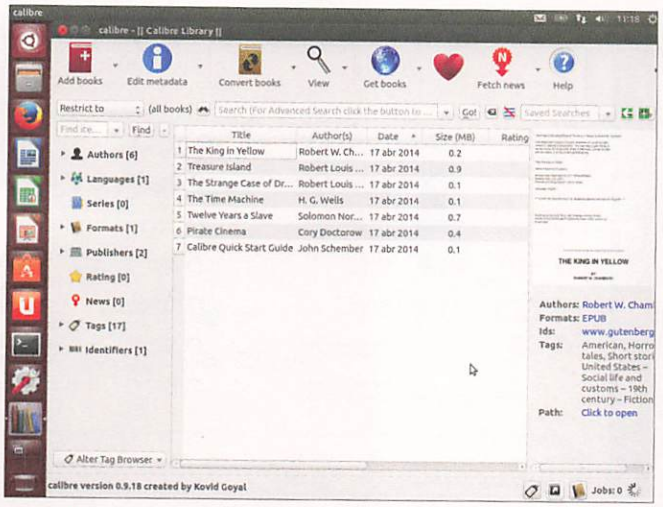


Figure 1: Clicking EPUB under the book's title image or double-clicking an entry in the library listing opens the reading window.

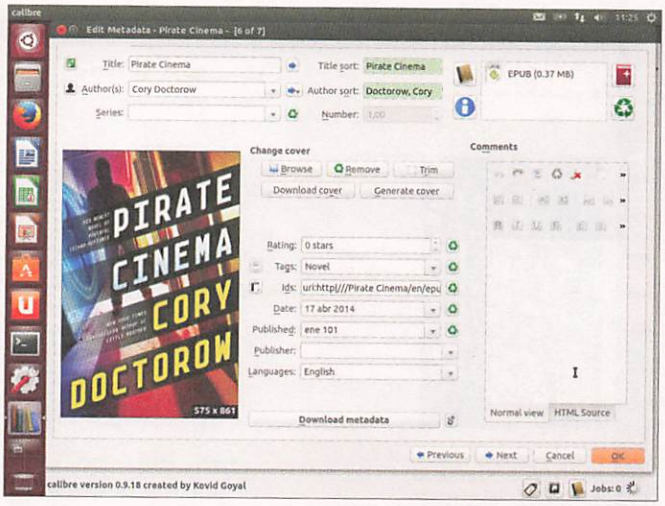


Figure 3: You can add or modify the metadata for a book.

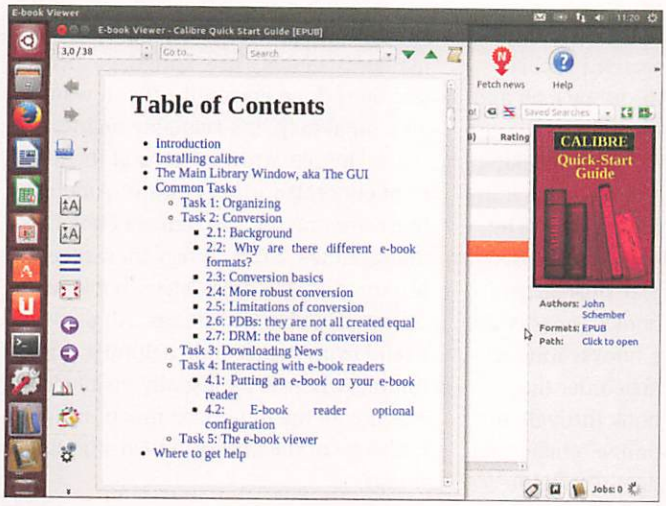


Figure 2: On the left side of the reading windows are the buttons for navigating and opening other views.

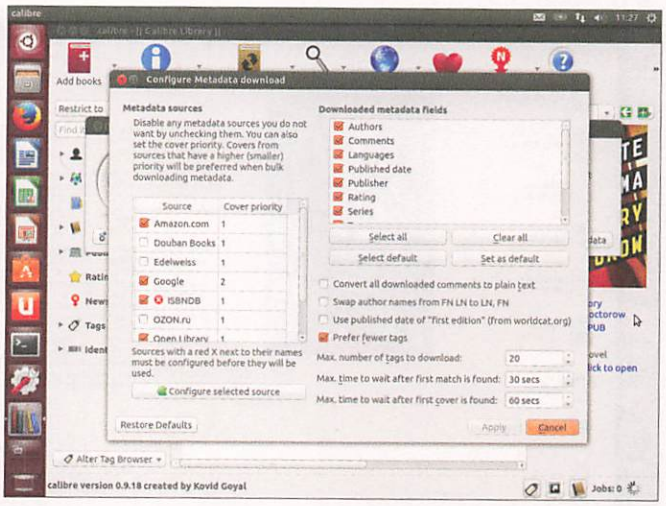


Figure 4: Instead of adding metadata for a book manually, you can easily extract it from one of many sources on the web.

pears in the library listing in the middle part of the window.

Taking time to add descriptions for new books is definitely worth the effort. To do this, you can click the *Edit meta information* icon at the top and choose the book from the list. Figure 3 shows the dialog with some incomplete data.

With the buttons to the left of the reading window, you set your preferred reading environment. The available print function somewhat defeats the purpose of the e-book, but at least it works.

Especially interesting is being able to extract metadata from the Internet. Press Ctrl + D and then *Configure Download* and identify the various metadata sources (Figure 4). After that, click *Download only metadata* (Figure 5) and you will see the added metadata.

Converting Books

The third button on the menu is for converting e-books. You can choose the conversion format from the list and set various parameters with the buttons on the left to control the complexity of the conversion process. Many of the settings for converting your books will be based on your own preferences and experience (Figure 6).

After you confirm your settings, the main window begins to show the conversion process visually. When it's done, the original, along with the converted copy, will appear in the directory with their respective file extensions to identify the formats. On the right of the Calibre main window, you can see both ver-

Installation

Many distributions provide installations for Calibre, but they're often outdated. Therefore, developers suggest copying and pasting a command that matches the distribution from the project homepage [1] to the terminal.

Although Calibre has no uninstaller, the authors guarantee that deleting the specified directories removes 99 percent of the project files. The developers also provide a command to update the software to the latest version or even revert to an older one.

Compiling Calibre from source proves to be complicated because of the many dependencies, and even the authors discourage it. Even so, the process is documented on the project homepage.

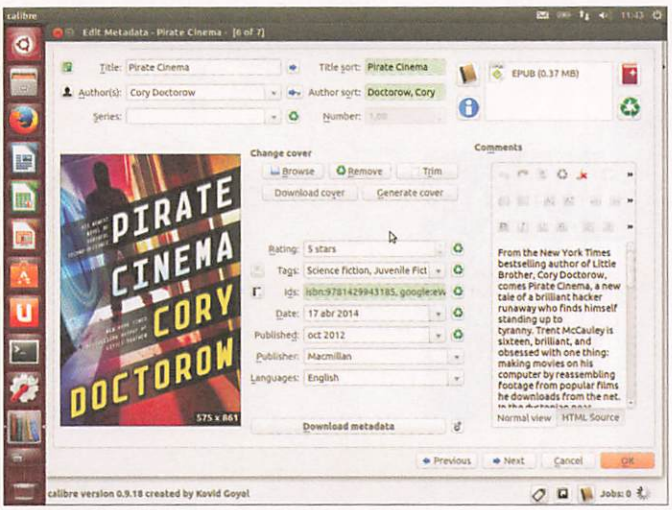

Edit Metadata - Pirate Cinema - [6 of 7]
Title: Pirate Cinema
Author(s): Cory Doctorow
Series:
Change cover
Browse
Remove
Trim
Download cover
Generate cover
Rating: 5 stars
Tags: Science fiction, Juvenile Fict
Date: 17 abr 2014
Published: oct 2012
Publisher: Macmillan
Languages: English
Comments
From the New York Times bestselling author of Little Brother, Cory Doctorow, comes Pirate Cinema, a new tale of a brilliant hacker runaway who finds himself standing up to tyranny. Trent McCauley is sixteen, brilliant, and obsessed with one thing: making movies on his computer by reassembling footage from popular films he downloads from the net.
Download metadata
Normal view
HTML Source
Previous
Next
Cancel
OK
calibre version 0.9.18 created by Kovid Goyal

Figure 5: When Calibre finds metadata on the web, you can easily add it for the book.

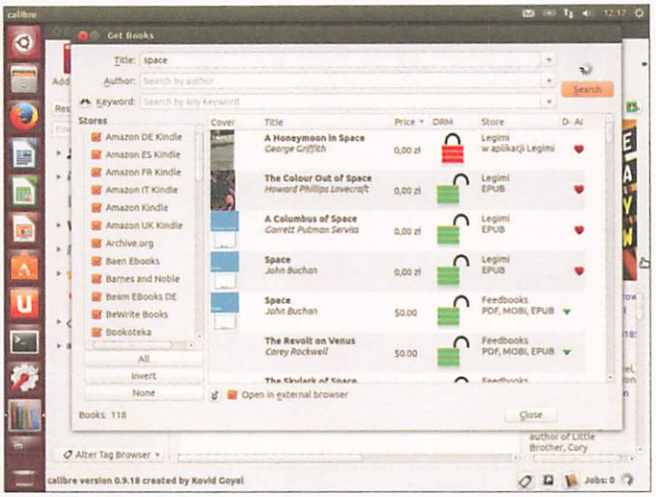


Figure 7: With Calibre, you can search the Internet for analog and digital books.

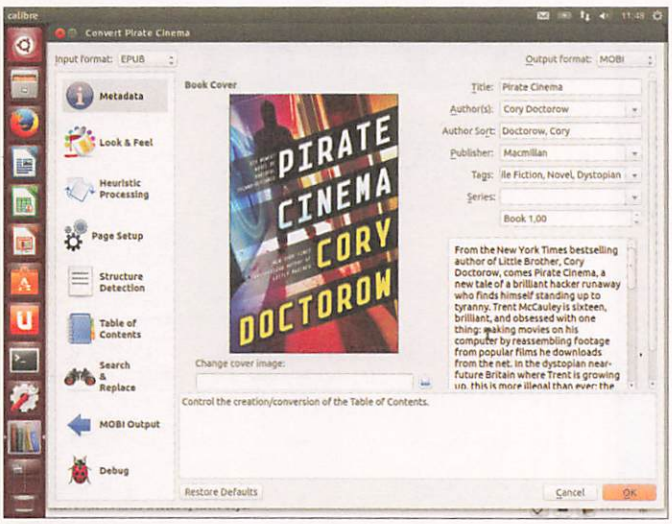

Convert Pirate Cinema
Input format: EPUB
Output format: MOBI
Metadata
Look & Feel
Heuristic Processing
Page Setup
Structure Detection
Table of Contents
Search & Replace
MOBI Output
Debug
Book Cover
Title: Pirate Cinema
Author(s): Cory Doctorow
Author Sort: Doctorow, Cory
Publisher: Macmillan
Tags: ile Fiction, Novel, Dystopian
Series:
Book 1.00
From the New York Times bestselling author of Little Brother, Cory Doctorow, comes Pirate Cinema, a new tale of a brilliant hacker runaway who finds himself standing up to tyranny. Trent McCauley is sixteen, brilliant, and obsessed with one thing: making movies on his computer by reassembling footage from popular films he downloads from the net. In the dystopian near-future Britain where Trent is growing
Change cover image:
Control the creation/conversion of the Table of Contents.
Restore Defaults
Cancel
OK

Figure 6: The form allows you to convert e-book formats, with numerous parameters to configure the process.

sions in their different formats, which you can then choose with a click. The conversion from EPUB [2] to MOBI [3] worked exceptionally well and without any error messages within seconds in my test.

The reading window, however, showed some significant differences between the two formats. The software did a better job with some of the document properties in EPUB format, perhaps because the heuristic rules worked out better. From a lack of some of the readers, however, I couldn't determine how the files might look on the respective hardware.

Searching

Calibre is able to search the entire web for books (Figure 7). You can use the title, author, or other keywords for the search. To start searching, click the globe icon and *Fetch books*.

Double-clicking a book opens it in the browser. The software can display normal print books as easily as e-books. You can also order the book through an online retailer and download it directly.

Searching for books for the first time can yield some overwhelming results. As always when searching over the Internet, the motto is "quality over quantity." In the case of Calibre, this means radically reducing the number of sources, or you end up with an untold numbers of hits on the same book from various vendors.

Affairs of the Heart

A special feature is the heart icon. Clicking it opens a webpage where you can donate to the Calibre project, which can help preserve the numerous features that make Calibre an excellent program for managing digital libraries.

A few places show where the developers may have gone a bit too far. For example, the *Fetch news* function transforms a news source webpage into an e-book. A feed reader would have been a more obvious choice for most users.

However, the program is largely self-explanatory in daily use. Language localization is done gracefully. If you want additional help, it's available online.

The double-arrow buttons at the top right conceals further options. Among these are functions to remove books, manage the Calibre library files, save the library to a hard drive, share books with others, and change settings. All these useful extra details hide behind a single button, whereas the usually visible icons execute all the important functions for daily use of the library, which simplifies things.

Conclusion

Calibre is a powerful library software that cleans up your e-book collection and, if desired, converts reading matter simply and thoroughly into various formats. In my test, Calibre even helped uncover and eliminate format errors in an existing e-book. To eliminate errors, however, you need to open an editor, such as LibreOffice combined with the eLAIX plugin.

Bookworms will appreciate being able to locate digital reading material on the Internet with Calibre. Of course, limiting the sources will save users from drowning in the sheer amount of all the possible hits.

Info

[1] Calibre: *http://calibre-ebook.com*

[2] EPUB: *http://en.wikipedia.org/wiki/EPUB*

[3] MOBI: *http://wiki.mobileread.com/wiki/MOBI*

Shop the Shop shop.linuxnewmedia.com

Want to subscribe?

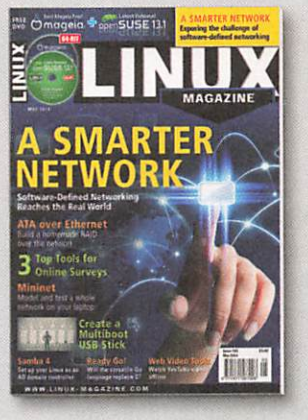
Need training?

Searching for that back issue you really wish you'd picked up at the newsstand?

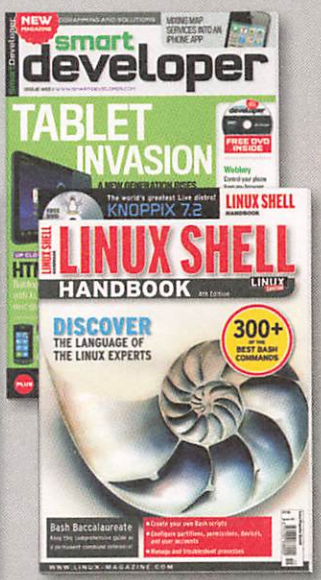
Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

➤ shop.linuxnewmedia.com

DIGITAL & PRINT SUBSCRIPTIONS



SPECIAL EDITIONS



TRAINING

- LPIC-1 LPI 101 - CompTIA Linux+ LX0-101
- LPIC-1 LPI 102 - CompTIA Linux+ LX0-102
- LPIC-1 - CompTIA Linux+ 101 + 102

"Where does all my money go?" A good personal finance management application like Buddi or HomeBank can help you to answer this question.

Buddi and HomeBank banking applications

Take It to the Bank

BY DMITRI POPOV

Money is a serious matter that deserves a serious tool to help you keep tabs on your income and expenses.

Why? Because tracking and managing your personal finances not only gives you an overview of your earnings and expenses, it also provides valuable insight into your spending habits over time. This insight, in turn, can help you manage your expenses and optimize your budget.

Jakub Krechowicz, 123RF

Buddi

Buddi [1] is a perfect tool for those who prefer to keep things simple. Although this application may look rather barebones, it provides basic tools for managing finances with a minimum of effort. As such, it is perfectly suited for users who are new to personal finance management.

Buddi, which is written in Java, requires a Java runtime environment. On Ubuntu, you can opt for the OpenJDK Java runtime environment, which can be deployed with the *sudo apt-get install openjdk-7-jre* command. To install Buddi, use the latest *.deb* package from the project's website.

Before you start using Buddi, you should configure its settings (Figure 1). By default, Buddi uses US dollars as the primary currency, so you might want to change that if you are not in the United States. To do this, choose *Edit | Preferences*, switch to the *Locale* section, and pick the desired item from the *Currency format* drop-down list.

If the currency you want is not included, you can add it manually by pressing the *Other* button and specifying the currency format. In the *Locale* section, you can also change the default interface language and data format.

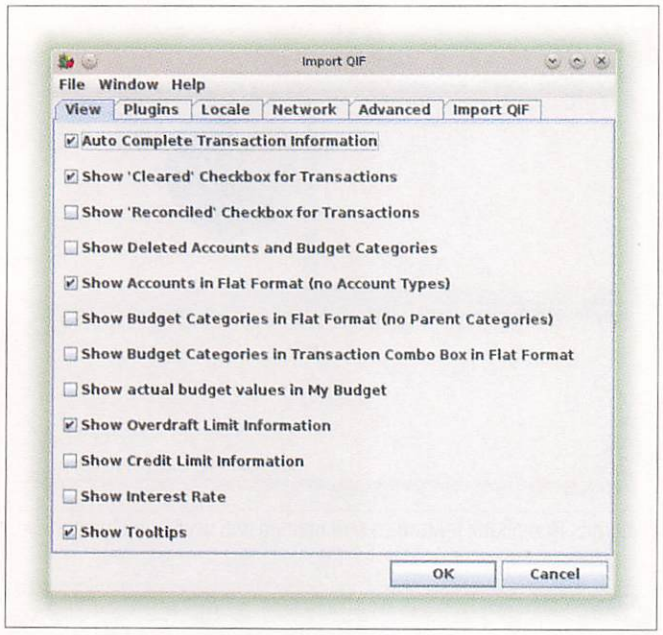


Figure 1: Configuring Buddi settings.

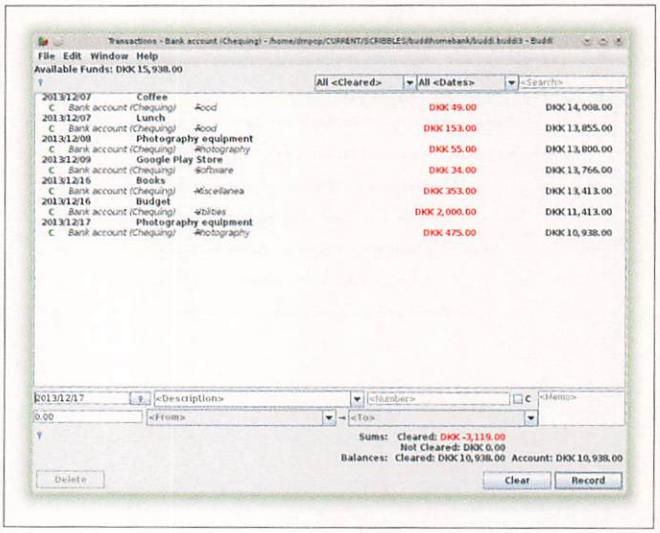


Figure 2: Working with transactions in Buddi.

The *View* section lets you enable and disable a few handy options. For example, you can choose to display accounts and categories either as a flat list or a hierarchical tree. You can also enable the *Cleared* and *Reconciled* check boxes, so you can mark transactions as cleared as well as reconcile them with bank statements. If your bank account allows overdraft, you can enable the *Show Overdraft Limit Information* option. Finally, in the *Advanced* section, you can tweak the auto-save settings.

Once you've configure the settings, you can start using Buddi. The first order of business is to add accounts and populate the application with categories. To do the former, switch to the *My Accounts* section, and choose *Edit | Create Account*. Give the account a name, choose its type from the *Account Type* drop-down list, enter the starting balance and overdraft limit (if available) values, then press *OK*.

When working with accounts in Buddi, keep in mind that the application operates with two account types: credit (accounts to which you owe money) and debit (accounts in which you have money). For example, cash and checking accounts belong to the debit type, whereas credit card and loan accounts fall into the credit category.

The next stop is the *My Budget* section, where you can specify budget categories. Using the commands under the *Edit* menu, you can delete the predefined categories you don't need as well as add new entries. Buddi also supports subcategories, which allow you to keep better track of your expenses. For example, you can create a category called *Books* containing several subcategories like *Fiction*, *Non-Fiction*, *Technical books*, etc. Later, you can see how much you spend on books in general as well on each type of book.

Next, you can populate Buddi with transactions. A transaction is a record of what you've spent money on, and which account you've used for purchase. For example, if you bought a cup of coffee and paid for it from your pocket, the transaction would be from your *Wallet* account to the *Food* budget category.

To add a new transaction as well as manage existing ones, double-click on the desired account. This step opens the *Transactions* window with a list of all recorded transactions for the current account. If no record is selected, you can create a new transaction by simply filling out the required fields at the bottom of the window (at the very least, you have to provide the date, amount, and sources); otherwise, you need to press the *New* button first. Press *Record* to save the transactions.

To modify an existing transaction, you simply select and edit it, then press the *Update* button (Figure 2). Buddi also supports so-called scheduled transactions, which can come in handy when you need to keep tabs on payments that occur on a regular basis, for example, donations, club membership payments, house rent, and so forth.

To create a scheduled transaction, choose *Edit | Edit Scheduled Transactions* in the main window. The only difference between a regular and scheduled transaction is that the latter occurs on a specified date, so you have to add a payment schedule for it. Once you've done that, Buddi will automatically add a record to the transaction list each time the scheduled transaction occurs.

Buddi supports reporting capabilities that can help you to make sense of all the entered data. Generating a report is as easy as choosing a time period from the appropriate drop-down list in the *My Reports* section (Figures 3 and 4). Using the available report types, you can generate pie charts of your income and expenses, display a list of all transactions, and show all income and expenses grouped by category, etc.

Finally, you can extend Buddi's functionality by installing plugins. The list of available modules is rather short, but it contains a few handy plugins [2]. For example, the Import QIF to import financial data in the QIF format from other applications into Buddi. To install the plugin, download and unpack it. In Buddi, choose *Edit | Preferences*, click on the *Plugins* tab, press the *Add* button, and select the resulting *.jar* file. Restart Buddi to activate the plugin.

HomeBank

Although HomeBank [3] is a step up from Buddi in terms of functionality, the

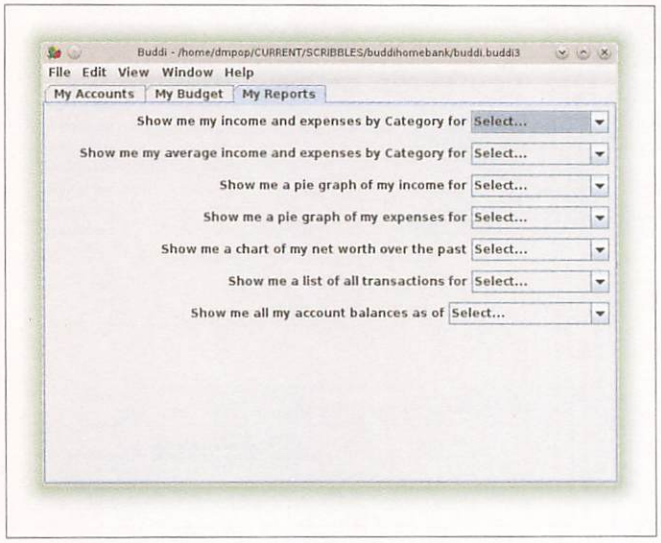

Buddi - /home/dmpop/CURRENT/SCRIBBLES/buddihomebank/buddi.buddi3
File Edit View Window Help
My Accounts
My Budget
My Reports
Show me my income and expenses by Category for Select...
Show me my average income and expenses by Category for Select...
Show me a pie graph of my income for Select...
Show me a pie graph of my expenses for Select...
Show me a chart of my net worth over the past Select...
Show me a list of all transactions for Select...
Show me all my account balances as of Select...

Figure 3: Buddi sports basic reporting capabilities…

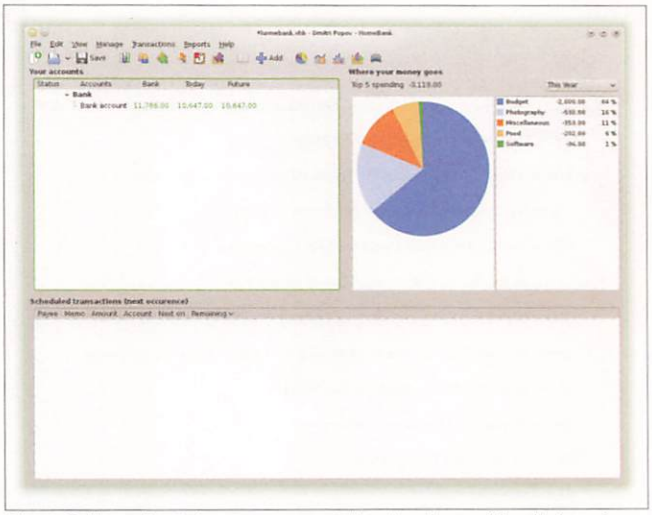


Figure 5: HomeBank features a streamlined and user-friendly interface.

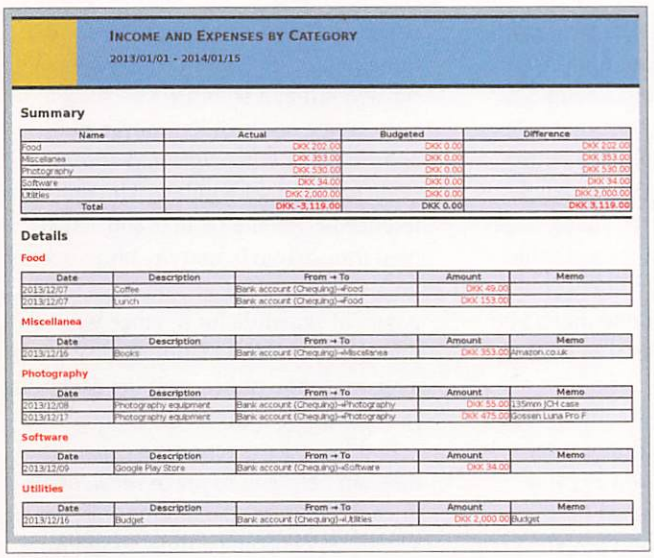


Figure 4: … and it generates reports in the HTML format.

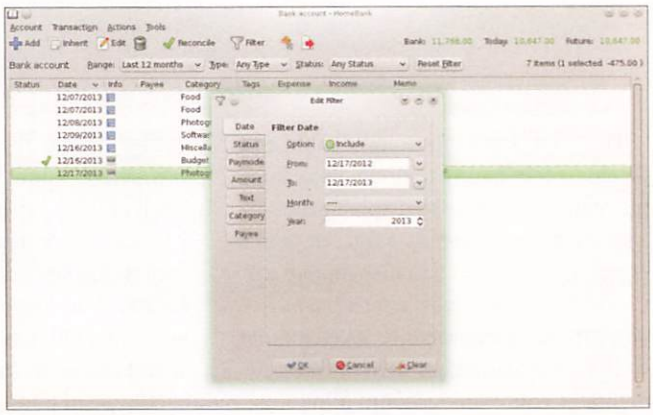


Figure 6: HomeBank boasts powerful filtering functionality.

former uses the same basic approach to managing personal finances as the latter (Figure 5). HomeBank has been around for some time, so the application has made its way to the official repositories of many popular distros, including Ubuntu. This means that you can install HomeBank either through the Software Center or by simply running the *sudo apt-get install homebank* command.

The first step is to create a wallet and populate it with accounts, payees, and categories. Choose *File* | *New* to create a new wallet. To add an account, choose *Manage* | *Accounts* and press the *Add* button. Give the account a name, then enter the account number and the name of the bank in the appropriate fields. Next, enter the current account balance in the *Initial* field. You can configure other options, if needed, and then press *Close* to add the account and close the window.

Next, you need to add payees, or people and companies on the sending or receiving end of a transaction. Choose *Manage* | *Payees* and use the *Add* button to add payees. Finally, you have to specify categories, which is done by choosing *Manage* | *Categories* and adding the desired options. When creating categories for your income, remember to tick the *Income* check box. In addition to accounts, payees, and categories, HomeBank allows you to specify and configure other useful items.

Assignments can be used to automatically assign a category and payee to transactions containing a certain text in the memo field. For example, you can create an assignment that automatically sets the payee to *Amazon* and category to *Passive income* when you add a transaction that has the *Kindle Direct Publishing* string in the memo field. Setting up an assignment is easy: Choose *Manage* | *Assignments*, press *Add*, specify the desired text string, payee and category, and you are done.

HomeBank also makes it possible to configure budgets, which can be handy for keeping certain expenses under control. For example, if you want to limit your book-related purchases to a certain monthly amount, you can set up a budget for the *Books* category. To do this, choose *Manage* | *Budget*, select the appropriate category, and enter the maximum allowed amount globally or for each month.

Of course, this feature won't actually prevent you from spending more on specific items than you budgeted for, but it provides a strong psychological incentive to stay within the specified limit. As any personal finance software worth its salt, HomeBank supports scheduled transactions and templates, which you can set up and configure by choosing *Manage* | *Schedule/Template*.

Once you've done the preparatory work, you can start adding transactions.

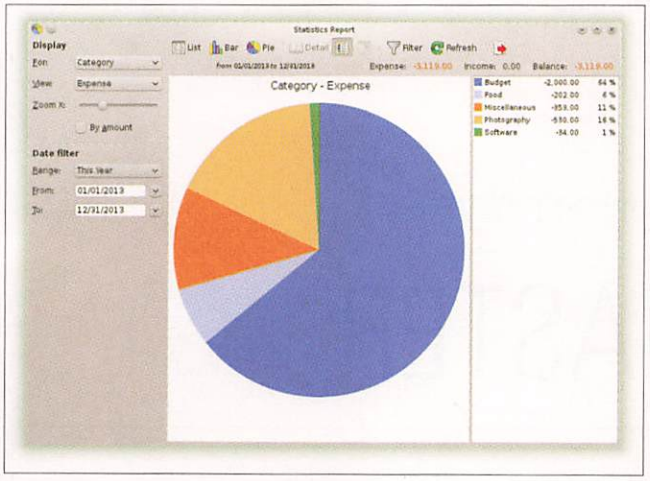


Figure 7: HomeBank lets you generate different types of reports.

Double-click on the desired account in the *Your accounts* pane and press the *Add* button in the main toolbar. Besides the standard fields, the transaction window contains two check boxes: *Reconciled* and *Remind*.

You can use the former to mark the transaction as confirmed after you've checked it against your bank statement. The *Reminder* check box can be used when you are adding a transaction for lending money. When you tick the check box, the transaction acts as a reminder, and it's not included in the balances.

The transaction interface provides several tools for working with transactions. Using the *Range*, *Type*, and *Status* drop-down lists, you can narrow the list of transactions to a specific period. The powerful *Filter* feature allows you to define rather sophisticated filters to display the transaction that matches specific criteria (Figure 6).

HomeBank features powerful reporting and charting capabilities which make it easier for you to analyze your financial data. The application supports several types of reports, including general statistics as well as balance and budget reports (Figure 7). The statistics report gives you the ability to view transactions by category and type, filter transactions by specific criteria, as well as view them as a bar or pie chart. As the name suggests, the balance report provides tools for analyzing your balance, whereas the budget report helps you get an overview of your running budget and see it as a bar chart.

Final Word

Buddi and HomeBank are not the most advanced applications for managing personal finances. However, they strike a good balance between usability and functionality. Using these applications, you can keep tabs on your income and expenses without being inundated with advanced features. •

Info

[1] Buddi: *buddi.digitalcave.ca*

[2] Buddi plugins: *buddi.digitalcave.ca/plugins.jsp*

[3] HomeBank: *homebank.free.fr*

Shop the Shop shop.linuxnewmedia.com

Missed an issue?

You're in luck.

Most back issues are still available. Order now before they're gone!

shop.linuxnewmedia.com

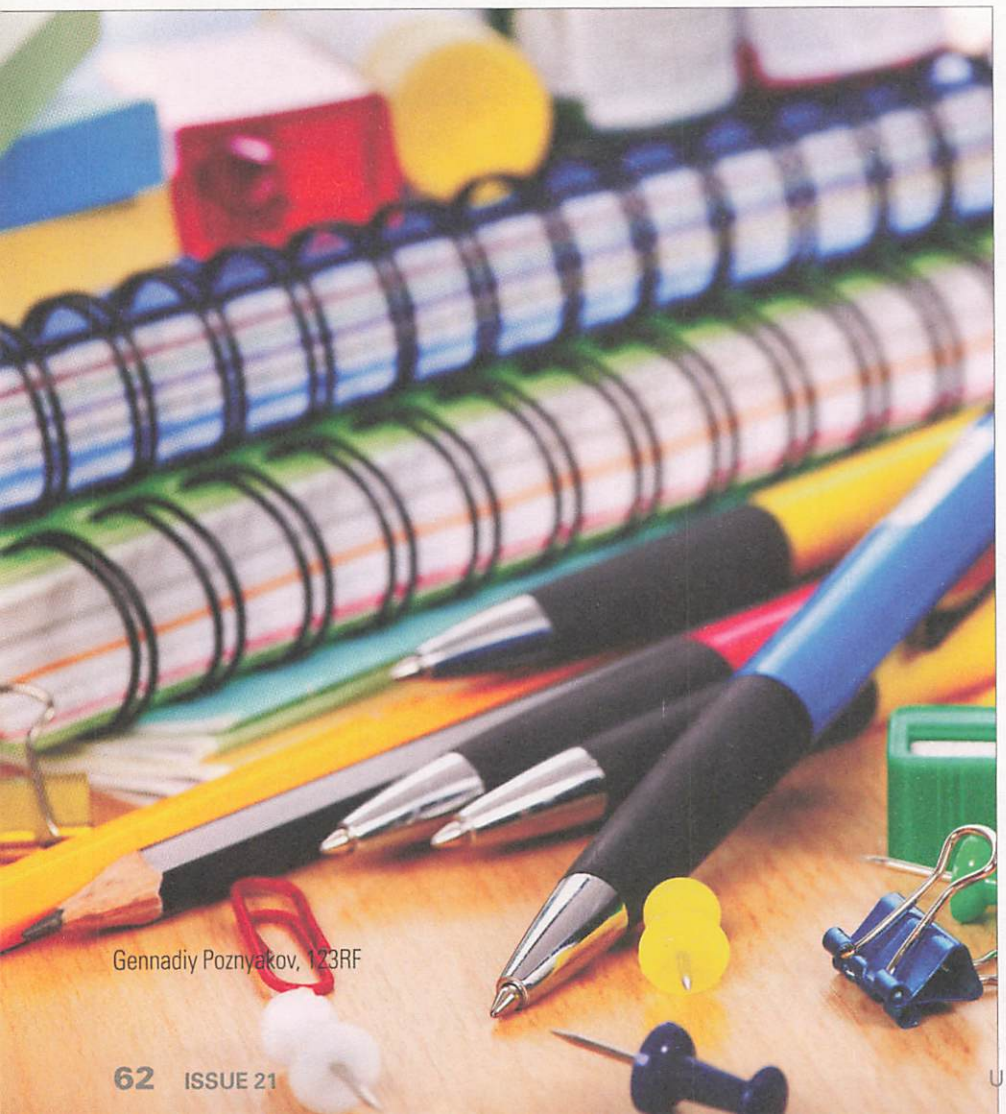
Nitro is an application with a modern design created for managing task lists. This free program even synchronizes with different systems.

Organize tasks professionally with Nitro

TASK MASTER

BY STEPHAN LAMPRECHT

The tastes and work approaches for task management tend to be quite different. One person simply needs a sheet of paper or a long document in a word processor to jot down what's important. Someone else might carry around a thick planner or use software with a fixed structure to keep track of activities and tasks.



Gennadiy Poznyakov, 123RF

Nitro [1] is a task manager that differs from other programs in many ways. A few years ago a new approach called "Getting Things Done" (GTD, [2]) began to win favor in task management.

This approach proposed categorizing tasks by solutions appropriate for them, or even by location, instead of assigning priorities to the tasks themselves. Thus, to get professional and private tasks under one roof, they need to go on separate lists.

Nitro supports this approach, without necessarily clinging to it. Whereas the GTD inventor eschews assigning priorities, Nitro permits prioritization along with setting deadlines.

Nitro is an exception among current software in that the program is available only as a standalone application for Linux and the Mac.

Windows users have to fall back on the appropriate extensions for the Google Chrome browser or manage their tasks exclusively on-

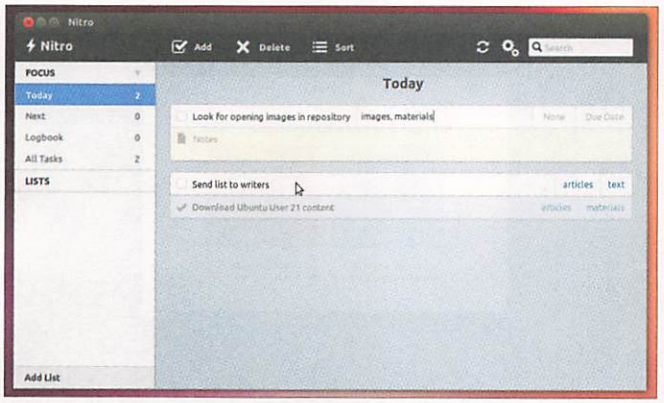


Figure 1: Nitro organizes tasks by priorities and tags.

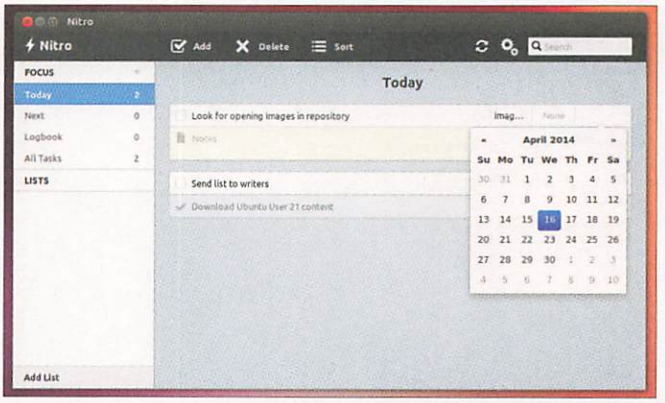


Figure 2: Nitro opens a calendar to choose an optional due date.

line through the browser. Those interested in the web interface don't need to be subjected to a long registration process; however: An account with Dropbox or Ubuntu One would suffice.

Installation

You have two ways to install the latest version of Nitro on Ubuntu. The easiest approach is to search for "Nitro" in the Software Center: The app is offered for purchase but is virtually free. The alternative to Software Center is that developers also provide the installation from a personal program archive (PPA) (Listing 1).

After installing the software, look on the Dash for *Nitro* and run the program. You'll initially get a blank program window. If necessary, change your language settings at this point on the *Language* tab by clicking the gear icon.

Unlike what you might be used to, the setting does not immediately take effect based on your choice. Exit the dialog and exit the program. When you restart Nitro, the language settings will be in effect.

Creating Tasks

Nitro has different methods for creating tasks. To add a template for a new task, for example, click *Add* on the icon bar or *File* | *Add Task* – or better yet, click Ctrl + N. Note the title or the actual task name. Experts agree that it's more motivating to use an active formulation for the title, such as "Add screenshot of Nitro task!" Press Tab to move to the next field.

Like many other task programs, Nitro supports organizing the tasks with tags or keywords (Figure 1). You can assign multiple tags, separated by commas, to a task.

Another Tab takes you a drop-down list of priorities to choose from, which you can choose with your mouse. And, another Tab takes you to a field to set a optional due date, where you can open a calendar and choose a date with the keyboard (Figure 2).

In the last field, *Note*, you can add an optional description – for example, more about the task and what you want to achieve by it. Finally, press Esc or click outside the dialog, and you can set up the next task.

Listing 1: Installing Nitro

```
$ sudo add-apt-repository ppa:cooperjona/
  nitrotasks
$ sudo apt-get update
$ sudo apt-get install nitrotasks
```

Lists

It wouldn't be clear just to store all the tasks in a simple series in the software. To get a clearer view, organize the tasks in lists. In *File* | *Add List* or by clicking *Add List* at the bottom left of the window, you can add an appropriate list name.

The way you structure tasks is up to you. You could distinguish between work and home tasks or create a list for each major project. Or, you can hold to the GTD philosophy and distinguish between resources and locations.

To add an entry to a list, you can click the list in the left navigation pane and click *Add* or use any of the other alternatives.

Views

With the help of the entries in the navigation pane, Nitro wants you to concentrate more on the essentials. One way is to navigate among the lists.

The optional *Focus* item at the top of the navigation pane groups the tasks without regard to lists. Thus, you can see which tasks you should perform based on their due date. *Next* shows you tasks that are due soon. To see all your tasks, click *All Tasks*.

One way motivate yourself is to occasionally visit the *Logbook* (Figure 3), which contains the tasks you've already completed – provided you add them to the logbook. If you add a check mark to a task or an entry in the Focus area, you mark it as completed.

Changing to the logbook shows you a note of how many items you've completed and which could be added to the logbook. Clicking the note adds the items to the logbook.

You can also switch focus from item to item using the *View* menu or by pressing Ctrl + 1 through 4 – with each number being the item in the navigation pane. To view *All Tasks*, for example, press Ctrl + 4.

Searching and Finding

The usefulness of a task manager very much depends on its search and sort functions. Users who ad-

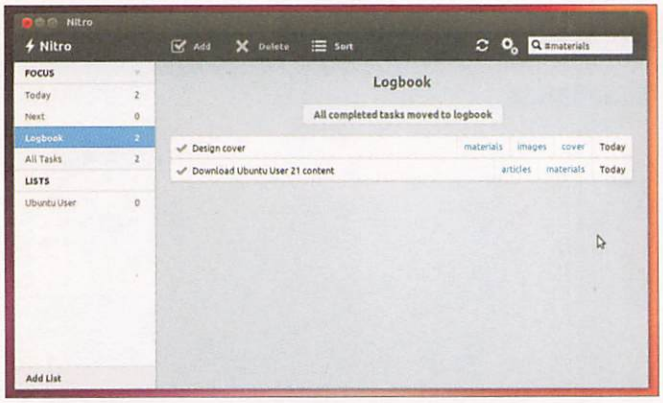


Figure 3: You save completed tasks in the logbook, which removes them from lists.

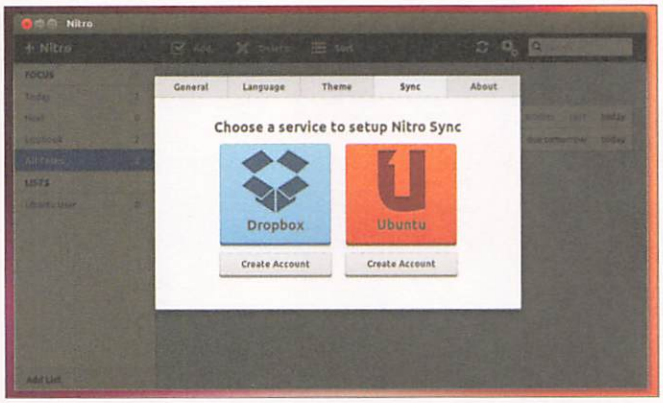


Figure 4: You can choose between two services to synchronize tasks between systems.

here to the GTD way of doing things in particular may have 100 or more tasks open at one time. The GTD approach is namely predicated on keeping all your work and private tasks listed at all times, to better clear your head.

In addition to organizing by lists, Nitro organizes by tags. The tags for a task are listed immediately after the title field. You can add tags by double-clicking on the task item. You can add multiple tags by separating them with commas.

Unfortunately, Nitro doesn't provide autocompletion of entries, so you explicitly need to enter the tags each time. One keystroke too many and you might create an unwanted extra tag, such as *Project* and *Projects*. A mouse-click brings up all the tasks with the same tag assigned. Clicking a tag in the list does the same thing.

The search function of the program is impressively fast. Press F to move focus to the Search field. Now you can enter a term or part of one. The search begins even while you type, and you start getting a hits with the first character alone. The search relates to the currently active list only, so it helps to have *All Tasks* showing before performing the search (Ctrl + 4 followed by F).

To search for an entry based on a day, you can prefix the search term with the pound sign (#). Often, however, you may not be so interested in a specific task or keyword but instead want to filter a list. Nitro developers thought of that and provided a sort function, which is available with a mouse-click on the icon bar.

In all, five sort options are available. You can sort by task title, due date, or priority. Two entries in the list differ from what's found in other applications. If you choose *Manual*, you can sort elements anyway you wish.

This approach is not very useful, however, because you can always change the search elements to another order with the mouse. Because of a bug, Nitro doesn't make the manual sorting persistent: If you move to another area and return, default sorting is restored.

A Nitro filter missing in other applications is the *Magical* sort, in which Nitro tries to group the open tasks through an algorithm based on tags, due dates, and priorities. Try it for yourself, and you're guaranteed new insights into the tasks.

Synchronization

No version of Nitro currently installs on a smartphone, so you will need a mobile browser. However, if you're working under different operating systems on different computers, you can set up synchronization among them. This approach works either through Dropbox or Ubuntu One. Unfortunately, the latter is being phased out by Canonical, so you won't be able to use it for much longer.

The setup is identical in both cases. Click the gear icon to open the settings and go to the *Sync* tab. Choose between the Dropbox and Ubuntu service to set up Nitro Sync (Figure 4). If you don't yet have an account for one of these services, Nitro can create one for you.

After making the choice, a little window opens so you can enter the URL for the service. Next, log in using your credentials. After you're in the system, you need to set up access to the service for the app (Figure 5).

Be sure to choose a name that you can remember. If you want to prevent access by external applications to the cloud storage later, it's good to recall which computer had the access.

Once the system accepts your authorization, you're prompted to close the window. At that point, you start the synchronization process. You need to repeat the authorization steps on each system where Nitro is installed. In the options, you can choose between manual synchronization or doing so at regular intervals.

New Looks

You can change Nitro's look in the settings. Simply click on the gear icon (or *View | Preferences*) and then on the *Theme* tab. Nitro provides various themes from which to choose. Without a doubt, the developers were inspired by the *Wunderlist* theme popular on Macs, which is why the eponymous theme exists in Nitro.

If you have any website design experience, Nitro lets you develop a completely unique look (Figure 6). Themes consist of graphics and

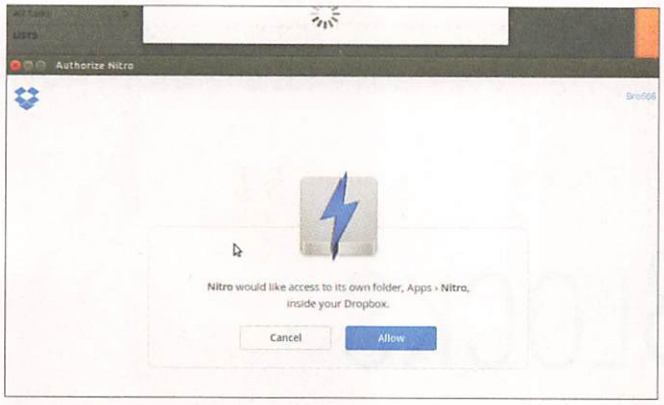


Figure 5: Both Dropbox and Ubuntu One ask whether you want to authorize access to the data, which you need to do on each system.

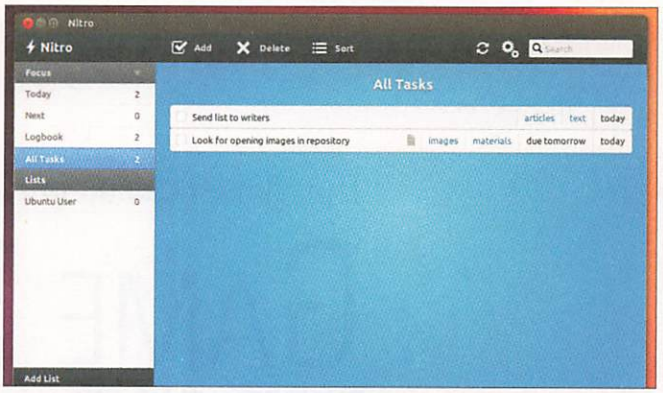


Figure 6: You can change Nitro's look with themes. If you know anything about creating CSS files, you can even create your own.

CSS files in the *opt/nitro/share/nitrotasks/media/app/css* directory that are basically templates. Always keep backups, though!

Conclusion

Nitro is missing a few features that keep it from being an all-around perfect task management system. It might not have occurred to developers that a user might want a printout of the task list – perhaps to carry around in a day planner. Yep, the PC program has no print function. You can only print from within the browser version.

You can access your tasks from anywhere, but the appearance may change from device to device. At the top of most users' wishlists is probably to have Nitro develop a mobile client for Android and iOS so that task lists are available on tablet PCs and smartphones. This functionality would be icing on the cake. •

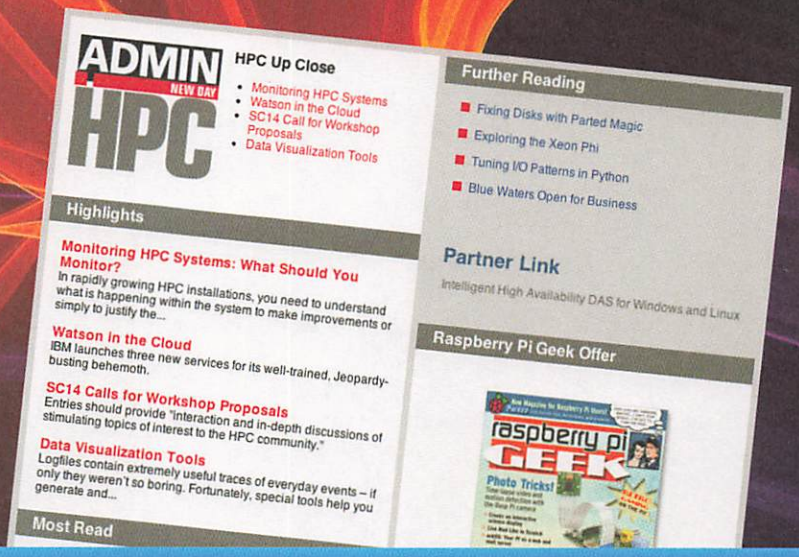
Info

[1] Nitro homepage: *http://nitrotasks.com*

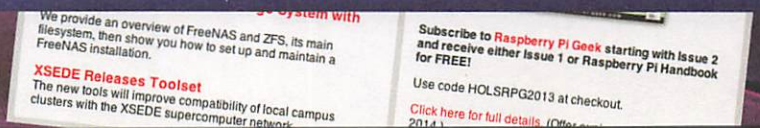
[2] GTD: *http://gettingthingsdone.com*

GOT CLUSTER?

Tune in to the HPC Update newsletter for news, views, and real-world technical articles on high-performance computing.


ADMIN
NEW DAY
HPC
HPC Up Close
Monitoring HPC Systems
Watson in the Cloud
SC14 Call for Workshop Proposals
Data Visualization Tools
Highlights
Monitoring HPC Systems: What Should You Monitor?
In rapidly growing HPC installations, you need to understand what is happening within the system to make improvements or simply to justify the...
Watson in the Cloud
IBM launches three new services for its well-trained, Jeopardy-busting behemoth.
SC14 Calls for Workshop Proposals
Entries should provide "interaction and in-depth discussions of stimulating topics of interest to the HPC community."
Data Visualization Tools
Logfiles contain extremely useful traces of everyday events – if only they weren't so boring. Fortunately, special tools help you generate and...
Most Read
Further Reading
Fixing Disks with Parted Magic
Exploring the Xeon Phi
Tuning I/O Patterns in Python
Blue Waters Open for Business
Partner Link
Intelligent High Availability DAS for Windows and Linux
Raspberry Pi Geek Offer
raspberry pi
GEEK
Photo Tricks!

hpc.admin-magazine.com/Newsletter


We provide an overview of FreeNAS and ZFS, its main filesystem, then show you how to set up and maintain a FreeNAS installation.
XSEDE Releases Toolset
The new tools will improve compatibility of local campus clusters with the XSEDE supercomputer network
Subscribe to Raspberry Pi Geek starting with Issue 2 and receive either Issue 1 or Raspberry Pi Handbook for FREE!
Use code HOLSRPG2013 at checkout.
Click here for full details.

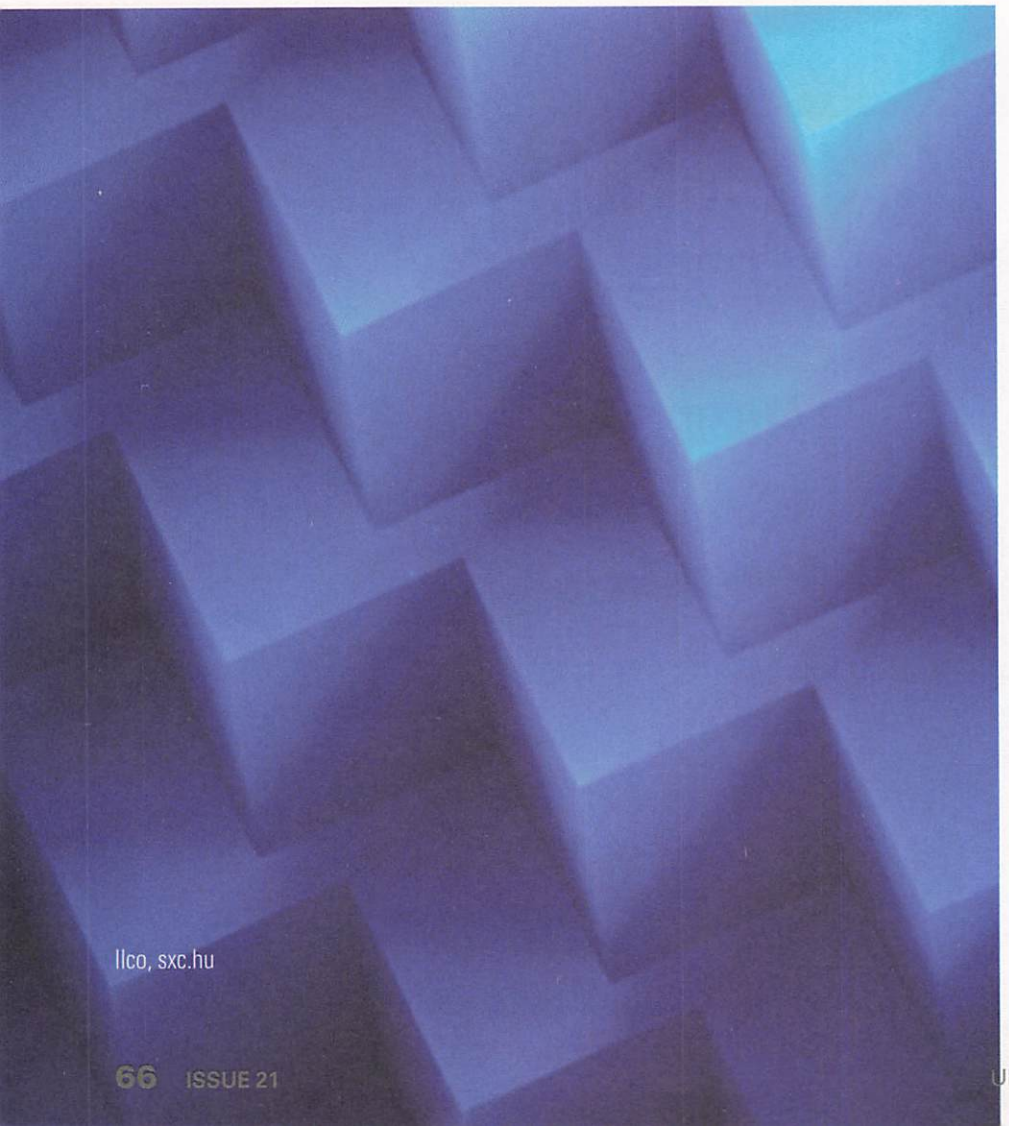
With the current version 1.7, Minecraft developers have dared to take a step away from the popular blocks look. However, the real thrill of the game is hardly its visual effects.

The Minecraft cult game

GAME BLOCKS

BY HARTMUT NOACK

A small selection of games has been and still is available for Linux. Recently, Steam for Linux has added an interesting series of indie titles; but, as far as the really big, popular games are concerned, in most cases, you still have to hope the Windows version somehow works with Wine.



Ilco, sxc.hu

A really laudable exception is the game Minecraft [1] from the Swedish developer Markus Persson [2] and his company, Mojang. It's completely based on Java and OpenGL, so it runs splendidly on Linux (Figure 1).

Moreover, Minecraft doesn't bother you with complicated copy protection mechanisms; instead, it works with a simple registration over the Internet. All in all, working with Minecraft is like working with any freely licensed software.

If you've been living under a rock for the past year and are wondering what kind of game it is, it belongs to the Sandboxed Open Worlds category. These types of games simulate a world in which you can move freely, erect structures, and change the landscape. Minecraft also has elements of adventure and strategy games, such as hostile monsters, growing plants, and competition with other players in multiplayer mode.

STAY LOOSE

The one-time fee you pay for Minecraft is limitless, even for the fre-

Figure 1: Minecraft 1.7.2 brings new landscapes – and it runs flawlessly on Kubuntu 13.04.

Figure 2: Mods extend Minecraft with countless extras. Here you can see a Japanese village from Millénaire with giant trees made with extra biomes in the background.

quent upgrades. The user can do with the software what he wants – except redistribute it. Thus, there's no problem experimenting with other developers' changes.

Besides the stable version, Mojang also provides access to beta and even alpha versions for free download. Registration (20 euros, payable by credit card or PayPal) allows login to all versions of the game.

All downloads are freely available; the game asks for the registered username only at startup. Without an Internet connection, Minecraft provides the *Play offline* option, which has no restrictions but precludes any multiplayer activity. Apart from that, some mods (i.e., modifications or extensions) use the player's login to save the game status.

The game provides a very open structured programming interface for external developers, so that hundreds of mods and "texture packs" already exist.

With these, you can change the landscape, substitute objects, or alter light and shadow effects. Examples range from simple enhancements of new tools or new animals, all the way to games within the game, such as the civilization simulator Millénaire (Figure 2).

Game without Borders

After installing the program (see the "Installation" box), you can start the game with *java -jar Minecraft.jar*. By default, the launcher also loads subsequent updates. As of version 1.6, the option exists of storing installations in profiles,

Installation

To install Minecraft, download the *Minecraft.jar* Java file from the website [3]. The script opens a dialog where you enter your registered username and password. Clicking *Play* starts the standard version of the game or a configurable profile in which you put older or beta versions or self-assembled extended variants.

At startup, the script downloads the actual software and saves it in *~/.minecraft*, along with the everything else Minecraft needs. Self-built worlds and game statuses are saved in the *Saves* directory. If Minecraft Forge Modloader is installed, you'll find a *mods* directory for mods.

As with some other cross-platform programs, Minecraft is Java software. The demands on memory and processor are moderate. On an IBM T60 Thinkpad with 1GB RAM, the game wasn't quite fluid. On a PC with Intel Core i2 Quad and 3GB RAM, the standard configuration ran passably, although major enhancements caused some noticeable spottiness.

On a Lenovo IdeaPad with Core i3 and 8GB RAM, Minecraft ran fluidly under KDE alongside dozens of other programs. Even here, however, significant delays in rendering windows occurred during active gaming in some cases. In terms of graphics, the game requires decent hardware and an activated hardware accelerator.

The website and various instructions recommend installing the Oracle runtime environments for Linux. In our test, the game run well on Kubuntu 13.04 with Open Java. Even with some of the bleeding-edge advanced packages, we did not detect any problems.

Figure 3: Lava exists in the upper and lower worlds, both in rectangular pools and flowing.

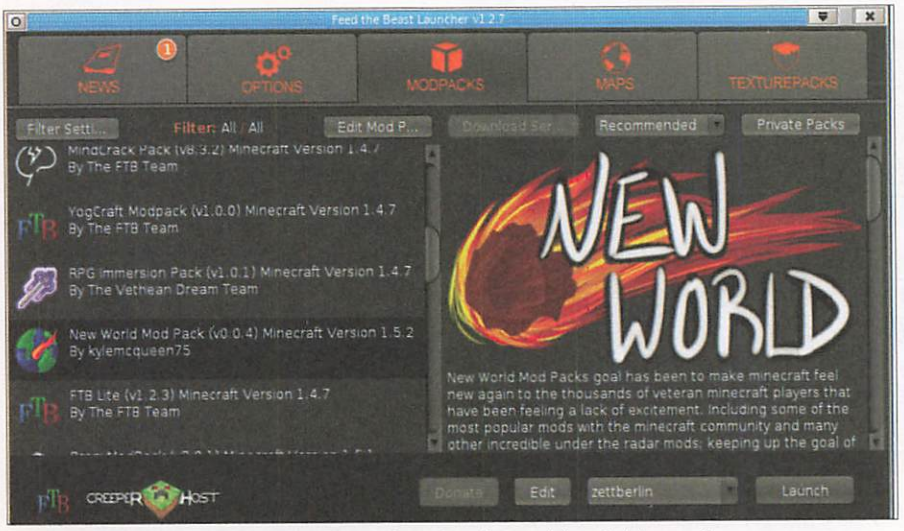

Feed the Beast Launcher V1.2.7
NEWS
OPTIONS
MODPACKS
MAPS
TEXTUREPACKS
Filter Setti...
Filter: All / All
Edit Mod P...
Recommended
Private Packs
MindCrack Pack (v8.3.2) Minecraft Version 1.4.7
By The FTB Team
YogCraft Modpack (v1.0.0) Minecraft Version 1.4.7
By The FTB Team
RPG Immersion Pack (v1.0.1) Minecraft Version 1.4.7
By The Vethean Dream Team
New World Mod Pack (v0.0.4) Minecraft Version 1.5.2
By kylemcqueen75
FTB Lite (v1.2.3) Minecraft Version 1.4.7
By The FTB Team
NEW WORLD
New World Mod Packs goal has been to make minecraft feel new again to the thousands of veteran minecraft players that have been feeling a lack of excitement. Including some of the most popular mods with the minecraft community and many other incredible under the radar mods, keeping up the goal of
CREEPER HOST
Donate
Edit
zettberlin
Launch

Figure 4: With "Feed the Beast," you can play using the graphics of older versions.

which don't automatically update the software.

The presets are not particularly conservative – all functions are turned on, including extras such as animated particles. Upon first startup, the installer also downloads the entire game (see also the "Installation" box).

Once you've applied a new world in single-player mode, the game character enters a landscape that looks different every time because of a sophisticated random terrain generator.

Minecraft includes landscapes such as jungles, savannahs, winter forests, and mountains made up of 10 different types of trees, some plants, and block-like objects. The blocks represent earth, sand, gravel, stone, and minerals.

Minecraft even renders snow, ice, fire, water, and lava from these rectangular blocks. Water and lava are rendered as animated liquids. Rain and snowfall provide for varied weather. There is no wind, however, and only a few simple sounds and a very muted soundtrack. The sound works perfectly, by the way, with Alsa as well as PulseAudio.

The software module that assembles the landscape is among the more remarkable technical achievements of Minecraft inventor Markus Persson. As the proportions of the game character suggest, the edge of a block is in reality roughly a meter in size.

With the character moving through the landscape, the module builds new elements without hitting any noticeable limit. A single world consists of a square with an edge length of a maximum 14 million blocks – the equivalent of five times larger than the earth.

The game places some animals in the landscape that you can hunt for raw material and food. Apart from pigs, chickens, and cows, horses have also grazed on the steppes since version 1.6, and they can be tamed and ridden. Less often, the game builds villages with virtual inhabitants with whom you can trade some goods.

The time of day changes every eight minutes. During those intervals, it's often advisable to seek refuge, because at night the game sends out a variety of monsters to hunt you down.

Even harsher is the underworld, known as Nether (Figure 3). You enter the Hell, elegantly rendered in dark reds, through a portal that you have to build yourself. For this and many other things, you need raw materials that you mainly procure through virtual mining – hence the name of the game.

Nothing is prescribed in the game. You can choose to live in a vegan pacifist mode, living entirely from plants and hiding at night to avoid fighting monsters.

Because the game invites exploration, but avoids bloody violence, it might be something parents of young children would appreciate – in Sweden, for example, the game is even used officially in classrooms. Without mods, the available weapons are limited to swords and a bow and arrows.

Creative mode is peaceful from the outset, with no concern for health and the option to fly through the countryside. It also is the easiest way to learn about the elements of the game, because there are no distractions of having to dig up or build blocks and devices; you can just simply extract them from an inventory list.

Gameplay Videos

A major part of Minecraft's success can be attributed the numerous videos on various websites, especially YouTube. Most of these videos show

LOST YOUR BOOKSTORE?

LET US BE YOUR BOOKSTORE

Browse our shop for single issues of *ADMIN*, *Linux Pro*, *Linux Magazine*, *Ubuntu User*, and *Raspberry Pi Geek* – delivered right to your door.

shop.linuxnewmedia.com/single

Better yet, subscribe, and you won't need a bookstore.

shop.linuxnewmedia.com/subs

shop.linuxnewmedia.com

DIGITAL AND PRINT EDITIONS AVAILABLE!

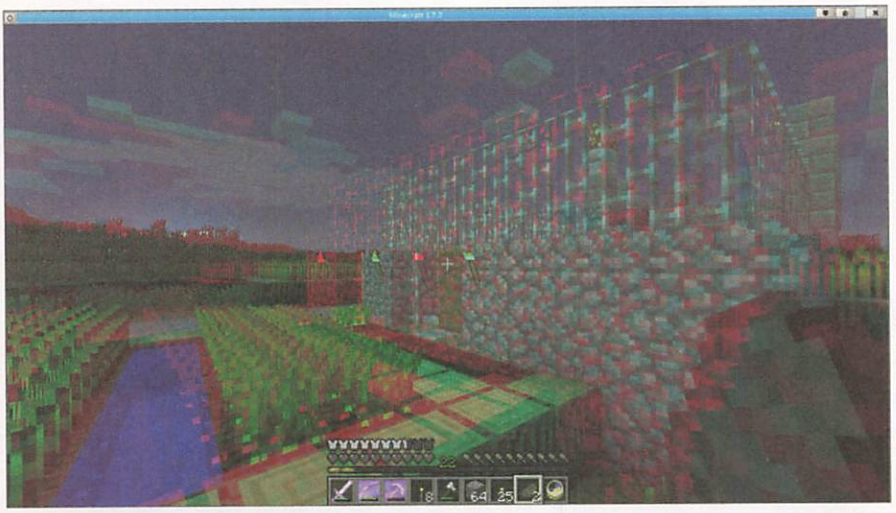


Figure 5: Not a misprint: Minecraft's 3-D mode provides a more realistic gaming experience with polarized glasses on.

a gameplay, which is more entertaining than it sounds. Many players and their teams are quite humorous and know how to stage exciting game situations or put together the "best of" from their games.

Players such as GommeHD [4], ZeronikHD [5], and Ungespielt [6], are all very entertaining players. The occasional slips of the tongue, pauses, repetitions, and similar technical errors give the videos an authentic charm, similar to professionally produced reality TV.

The videos also provide a kind of handbook. Some of the videos aimed at sharing experiences include breathtaking tricks. A player named Salaja built a 16-bit computer into his game world with help from five simple electrical elements of Minecraft. Salaja maintains a whole list of such installations in the Minecraft forum [7].

A few filmmakers use the game to make animated films. These range from "movies" of all genres created directly in the game, where live actors take on the roles as characters and monsters, to rather surrealist animations that play with the square aesthetics of Minecraft. Some of the small underground productions include astounding special effects in addition to being witty and original.

The works of CorridorDigital [8] include actors in grotesque papier mache costumes fighting zombies and hell monsters in some five-minute video clips.

The filmmakers incorporate 3D animations of Minecraft elements into the real world: Explosions tearing holes in the ground merge with the typical Minecraft blocks.

Open Landscapes

Multi-player mode provides the option to merge the actions and game statuses of multiple players in a virtual world on one server. On the download page of CraftBukkit, you can find a JAR file [9] that you simply start. In the wizard to the game, you enter the name or IP address of the server to which Minecraft automatically connects.

A prerequisite, however, is that all versions match exactly. If you operate a server, every Minecraft upgrade also includes an update for Bukkit, which may lock out older versions of the game.

Feed the Beast

If you are looking for a quick overview of the world of extensions for Minecraft, try out the mod package "Feed the Beast" (Figure 4).

The Java file *FTB_Launcher.jar* on the project website [10] does not install mod packages into the actual Minecraft installation. Rather, it installs complete systems for each package set, which allows you to play them separately.

The Blocky Future

The Minecraft game is continuously evolving. Version 1.7, for example, provides a few new landscapes, plans, and raw materials, and it shows some small steps toward modern graphics.

Sunsets are more dramatic, and the options include a *Supersecret Settings* button that allows you to turn on very experimental lighting effects. Additionally, you can try out a 3D mode, however, this mode requires you to wear polarized glasses (Figure 5).

The community around Minecraft receives these developments positively, but the simple graphics together with the open concept are really the main charms of this program. The aesthetics of block composition not only make Minecraft unique, but also contribute to its attractiveness.

Additionally, if you think Minecraft has become "too realistic" at some point, the open culture that has grown up around the game also provides "Primitive Blocks." This particular mod will restore Minecraft to its look from the good old days. •

Info

[1] Minecraft: *https://minecraft.net*

[2] Markus Persson homepage: *https://mojang.com/notch/*

[3] Downloading Minecraft: *https://minecraft.net/download*

[4] GommeHD: *https://www.youtube.com/user/GommeHD*

[5] ZeronikHD: *https://www.youtube.com/user/ZeronikHD*

[6] Ungespielt: *https://www.youtube.com/user/ungespielt*

[7] List of computers in Minecraft: *http://www.minecraftforum.net/topic/268582-list-of-minecraft-alus-cpus-and-computers/*

[8] CorridorDigital: *https://www.youtube.com/user/CorridorDigital*

[9] CraftBukkit: *http://dl.bukkit.org/downloads/craftbukkit/*

[10] Feed the Beast: *http://feed-the-beast.com*

QupZilla is a new Linux web browser that enters in an already crowded arena. But, QupZilla holds its own when it comes to speed and ergonomics.

The very fast QupZilla web browser put to the test

Speedy Newcomer

BY ERIK BAERWALDT

Apart from the usual top dogs, Firefox and Chromium, an overwhelming number of web browsers exist for Linux.

Kirill Zdorov, Fotolia

Some of them satisfy rather unusual needs, such as pure text representation of sites, which is what Lynx and ELinks are all about. If you find Firefox too bloated and Chrome too intrusive, however, give the newcomer QupZilla a whirl.

Although the QupZilla project is a mere three years old, binary packages are already available in many software repositories. You can install the browser with a few clicks in Mageia, Fedora, openSUSE, Debian, Ubuntu, and Arch Linux from the official software repository. The current version is available precompiled on the project website.

After a successful install, you can launch the browser from the HUD. QupZilla starts up quickly and leads you to an unobtrusive startup screen reminiscent of Firefox; however, it relies on the DuckDuckGo search engine instead of NSA-friendly Google.

The enormous speed of the browser derives from the Webkit rendering engine that allows fluid surfing even on less powerful hardware and eliminates stuttering when scrolling.

QupZilla's external look and user experience are closely modeled on Firefox, and it fits seamlessly into any existing work surface because of its Qt library base. Newcomers require no further training.

QupZilla integrates and activates the Adblock ad blocker that needs to be configured in a somewhat unusual dialog. QupZilla then sets the browser into a private mode, when a new window opens while the original one remains. QupZilla also uses the well-known tile (here called *Speed Dial*) view of Opera and Firefox that presents the recently visited web pages as miniatures on new tabs so that you can quickly visit them again.

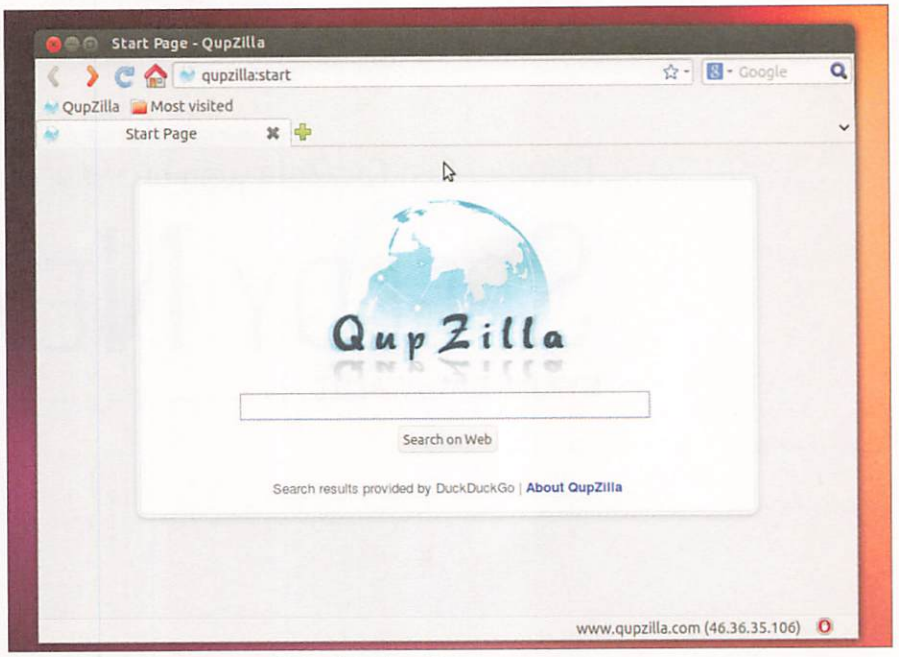


Figure 1: QupZilla opens up with its default settings.

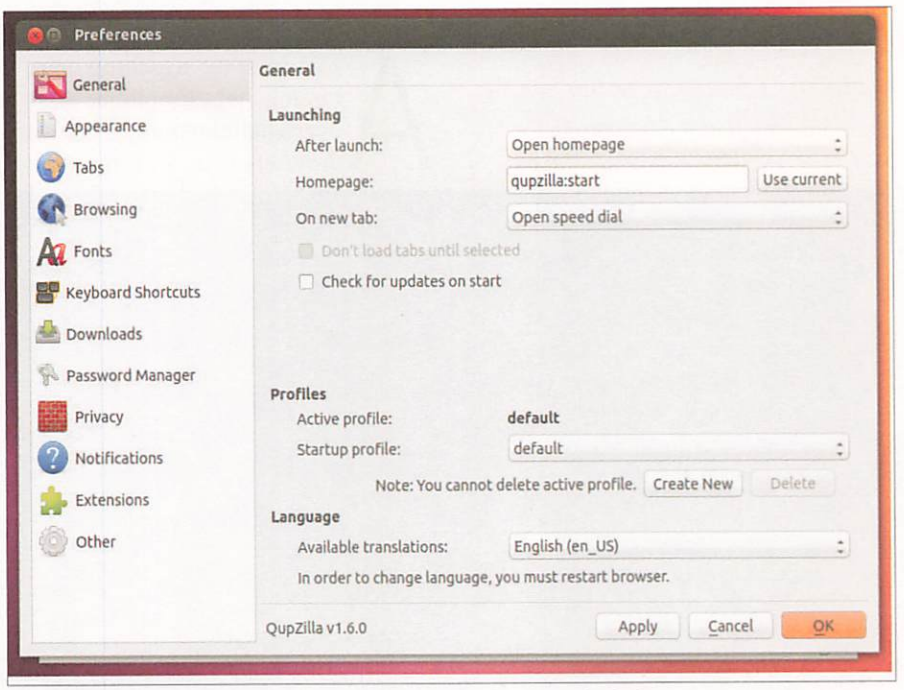


Figure 2: You can configure your browser in one clearly arranged window.

Settings

Despite its similarities to Firefox, QupZilla has a settings menu that deviate from Firefox quite a bit. You get to the settings from the horizontal menu bar with *Edit* | *Preferences*. A new window appears with several options in the left pane to customize your browser.

In the larger pane on the right, you can actualize the settings. QupZilla then partially groups the setting options in the right pane into horizontally arranged tabs (Figure 2).

The basic browser settings – such as home page, web configuration with JavaScript, cache size, download and password management, and privacy options – are made in the various subgroups of the settings menu. The options provided are similar to those in Firefox.

In QupZilla, the subgroup *Appearance* includes a selection of five themes with which to customize the browser. To activate one of them, click it and confirm your choice with the *Use* button at the lower right. QupZilla then changes the browser appearance immediately without a restart.

With the *Other* preferences, you can freely configure the icon and menu bars so that the browser can be in more of a kiosk mode with limited controls visible. Additionally, you can set the tab and address bar configurations just as freely from the *Tab* submenu.

Content

QupZilla allows you to hide certain resource-demanding content, especially for slower computers, such as Flash animation. In the *Preferences* | *Other* window, go to the *WebKit Plug-ins* tab and enable the *Allow Quick to Flash*, where you can define a whitelist of unblocked Flash addresses.

Similar to Firefox's widely used Flashblock add-on, QupZilla then shows only Flash videos in a frame from websites with the Flash icon. To view the contents, click the icon in the frame and only then will QupZilla load the data.

If QupZilla happens to crash during a session, it tries to recover all the previous tabs. You'll get a message with a list of recently opened tabs, and you can choose the ones you want to recover by entering a check mark.

Under a Pseudonym

If you notice that what QupZilla displays on a website varies from what you have seen using another browser, you can change the browser's user agent header. A single user agent header ensures a uniform content display for the web server. In the *Preferences* | *Other* menu, click the *User Agent Manager* tab.

In the ensuing window, add a check mark in front of *Change Global Settings* option. You open a list of predefined headers when you click the small triangle icon. From this list, you can choose a matching user agent parameter that the browser then sends to visited web pages (Figure 3).

To give separate web pages different user agent strings when calling them up, combine the web page and user agents in the entry field underneath, which ensures optimal viewing results. You only need to activate the *Use different user agents for certain websites* option with a check mark and enter the desired combination.

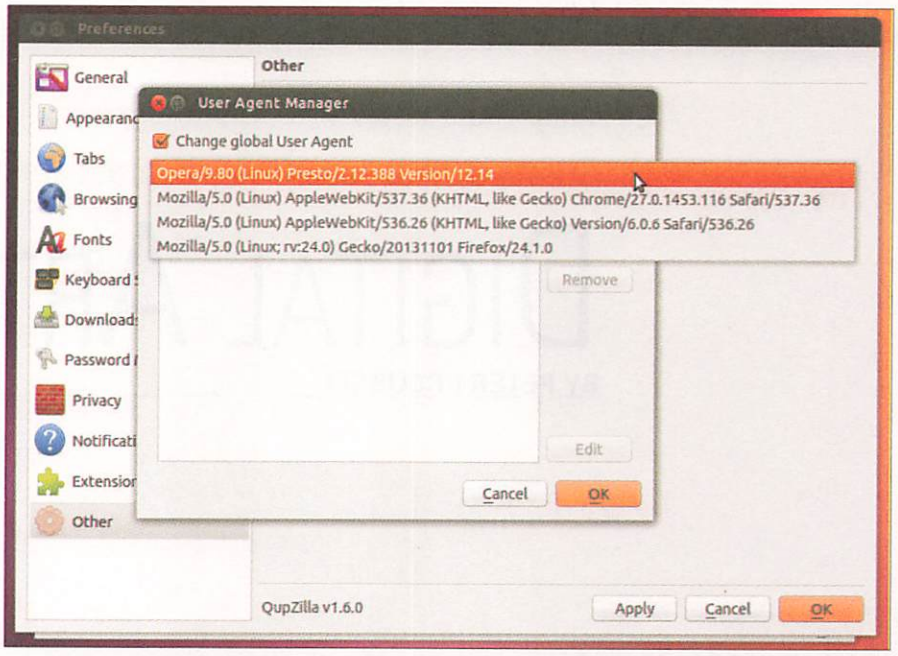


Figure 3: Using a single user agent header ensures uniform content display.

Bookmarks

QupZilla handles bookmarks for speeding up access to frequently visited web pages. You can also import previously defined bookmarks from Firefox. From the *Bookmarks | Edit bookmarks* menu, click the *Import Bookmarks* link on the lower right. In the ensuing window, choose whether you want to import the bookmarks from Opera, Firefox, Chromium, or an HTML file.

Next, enter the path to the original bookmark file, after which QupZilla pops up a notification depending on the browser. If you get an error message on an import that points to problems with reading in the bookmark database, it generally means that the original browser is still running and locking the database.

After successfully integrating the bookmarks, they now appear as usual in the *Bookmarks* menu (Figure 4).

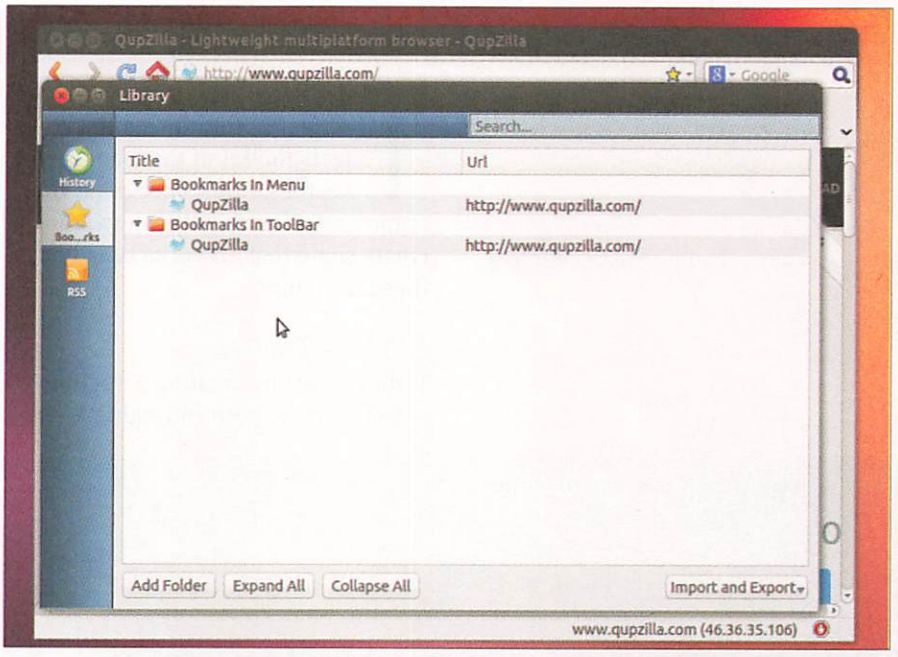


Figure 4: You can import bookmarks from other browsers.

Security

Unlike Firefox, QupZilla has not yet reached a level of security that covers all conceivable areas. But, thanks to its modular design with expansion capabilities via plugins, it's probably only a matter of time that currently open, albeit unknown, vulnerabilities get patched.

QupZilla blocks advertising by default and provides powerful certificate and cookie management. It also sends do-not-track headers on request. Still missing are extensions for blocking web bugs and deleting so-called "local shared object" (LSO) cookies. Adobe Flashplayer stores user-specific data in such Flash cookies for later retrieval by a restored website or application.

To block undesired content from the outset, QupZilla provides the option of an upstream proxy server. The corresponding dialog is in *Preferences | Browsing | Proxy Configuration* so that you can set the proxy service such as you would, for example, in Firefox.

Conclusion

QupZilla is a promising newcomer to join the ranks of the established web browsers. Externally similar to Firefox, it is stable, standards-compliant, and also very fast.

Thanks to its modular design, QupZilla can be upgraded with enhancements. Only in terms of safety is there some catching up to do. Extensions for blocking web bugs and removing Flash cookies are still missing. •

Info

[1] The QupZilla web browser: *http://www.qupzilla.com*

The Inkscape vector graphics program not only lets you work with pencil and ink lines, it can help you mimic the elegant brushstrokes of an oil painting. Our three easy-to-follow examples show you how.

Creative Inkscape effects in practice

DIGITAL ART

BY PETER KREUSSEL

The first article of this two-part series [2] introduced the effects toolkit of the leading free vector graphics program, Inkscape. In this article, I'll show how to create some real artworks based on these techniques.

GETTING STARTED

You can start by creating a slightly oriental, shiny pattern (Figure 1, bottom right), which is quite easy to do in Inkscape. For the intricate pattern, you can use a photo from a Persian wall mosaic [3] (Figure 1, top left), which you can trace with Inkscape's vectorizing tool and create the image you see in Figure 1, top right.

This image would end up having too many nodes to manipulate comfortably, so you should reduce it in advance to 1,000 pixels wide.

Nailia Schwarz, 123RF

PEELING AWAY

The vectorization function *Path | Trace Bitmap* transforms the golden arabesque pattern in a few mouse clicks to a vector format (Figure 2). Recognizing shapes in photos is a challenge for computers, even if it is easy for humans. Nevertheless, with a proper use of Inkscape's vectorization tool, you can get very good results.

Luckily, the original Persian wall mosaic photo, despite its delicacy, is easy to convert, because of the contrast between the blue background and yellow pattern. Therefore, you would not need the *Edge detection* algorithm (not selected in Figure 2) that finds the edges in images with many similarly colored elements.

The default tracing mode *Brightness cutoff* with a value of *0.45* is

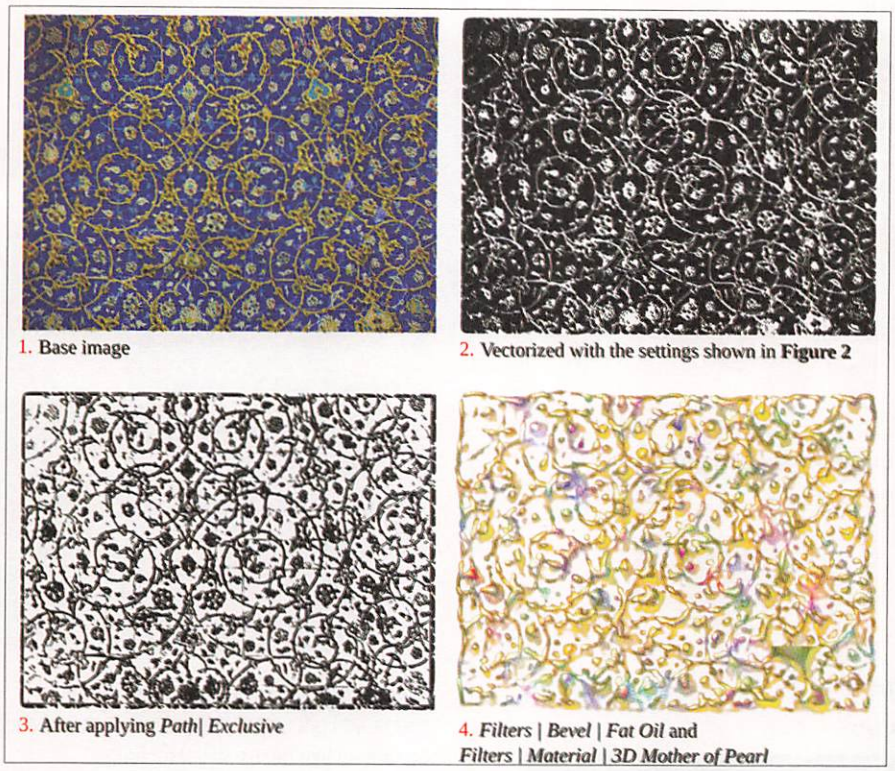

1. Base image
2. Vectorized with the settings shown in Figure 2
3. After applying Path| Exclusive
4. Filters | Bevel | Fat Oil and Filters | Material | 3D Mother of Pearl

Figure 1: From an original photo (1), you can apply the settings shown in Figure 2 to create a vectorized version (2). Then, apply an exclusive-or function (3) and finish with "fat oil" and "mother-of-pearl" effects (4).

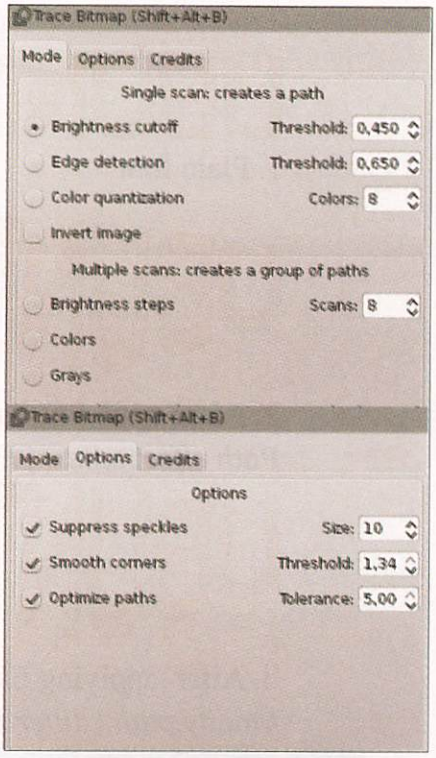


Figure 2: You can use the settings in Path | Trace Bitmap to vectorize the Persian wall mosaic.

enough to detect the yellow mosaic elements. The turquoise-colored element is considered dark, but that actually works well to give the result (Figure 1, top right) an uncluttered look.

This kind of vectorization renders the image in black-and-white. The Inkscape handbook describes how to generate multicolored images [4].

The complexity of the paths created through vectorization presents one of the challenges of detecting bitmaps: The number of nodes can get so high that it can take Inkscape minutes to display them.

You can control this through the *Options* tab in *Trace bitmap* (Figure 2, bottom) by setting the *Suppress speckles* option for the specified *Size*, which you can set to 10 pixels, for example.

For *Smooth corners* and *Optimize paths*, you should choose the maximum values *1.34* and *5.00*, respectively, for the smoothest possible yet not too complex result.

Mother of Pearl

The vector transformation should then become a negative image for the final patterning that is to follow. You can check off *Invert image* on the *Mode* tab to turn it to positive again. Unfortunately, this step renders the image visually compact with a dark edges.

A better solution is to draw a filled rectangle that covers the pattern but leaves a small border. Then, use *Path | Exclusion* to invert the area covered by the rectangle, which leaves the narrow dark strip around the edges of the original negative.

The powerful Inkscape filter effects handle the rest. *Filter Effects | Bevels | Fat Oil* coats the flat, woodcut-like forms with a viscous oil layer to give it an almost 3D shading effect. *Filter Effects | Materials | 3D mother of pearl* adds an iridescent frosting to the pattern.

Letter Acrobatics

In elaborate graphic design applications, you can get text to flow on more than just straight lines. One approach to playing with text is found in the Inkscape function *Text | Put on Path*; however, this tool is somewhat too reminiscent of the word-art function of word processing programs and is equally limiting. To shape text really creatively, you must convert it into vector shapes. Then, the rich palette of Inkscape path effects becomes available to satisfy your creative imagination (Figure 3).

You first convert the text to a vector form with *Path | Object to Path*, which puts the text into a set of paths to which, unfortunately, you can't apply any path effects. Therefore, you can divide up the group with Ctrl + U and recombine the individual text character objects into a single path with Ctrl + K.

You can then "knead" the path with *Path | Path Effect Editor | Envelope Deformation*. This tool makes the text-manipulation utilities of your typical word processors look downright primitive in comparison. With *Envelope Deformation*, you can stretch the four corner control points (nodes) into Bézier curves with the *Node Tool* (Figure 3, second diagram). You can manipulate these so-called envelope curves in the *Path Effect Editor* (see Figure 4).

Despite the creative deformation of the whole object, the outlines of the text still look too smooth. To help

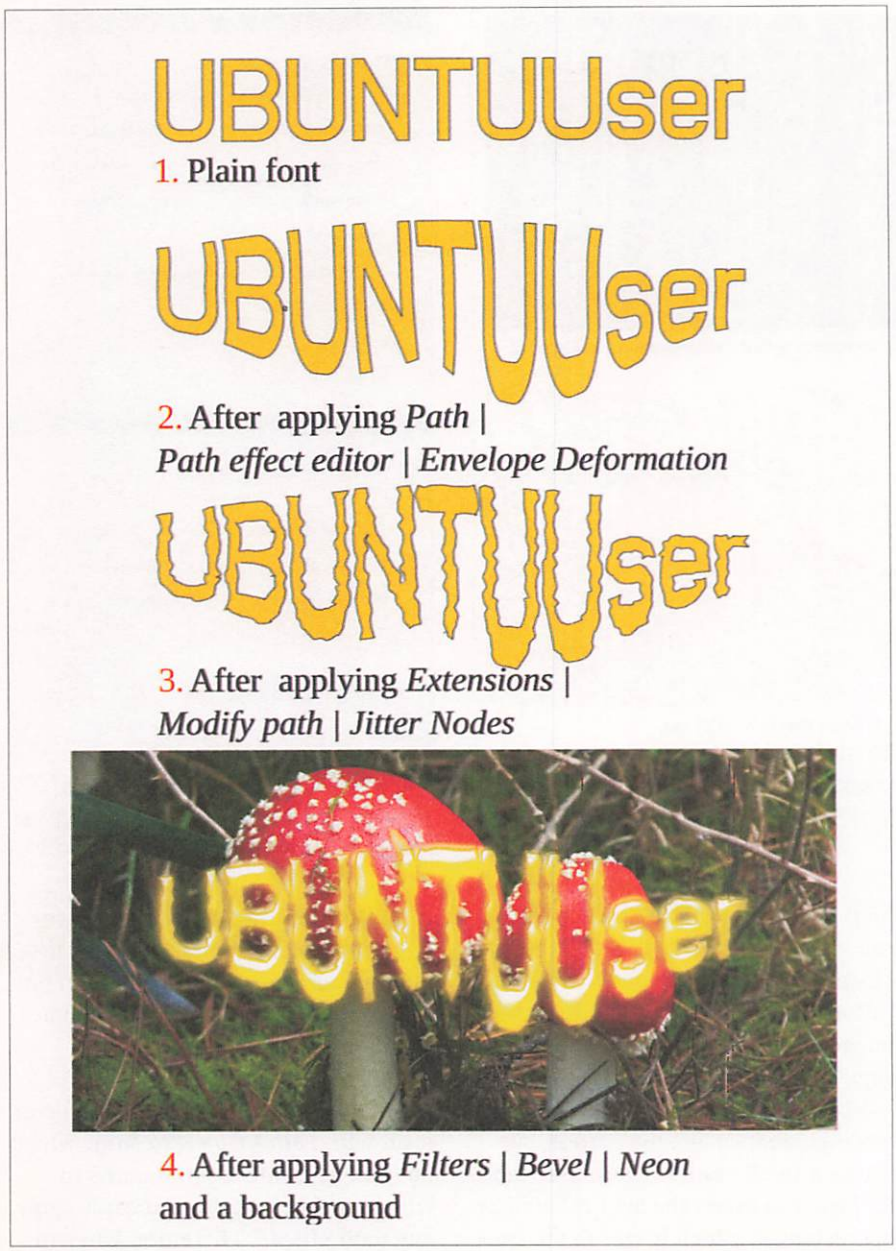

UBUNTUUser
1. Plain font
UBUNTUUser
2. After applying Path | Path effect editor | Envelope Deformation
UBUNTUUser
3. After applying Extensions | Modify path | Jitter Nodes
UBUNTUUser
4. After applying Filters | Bevel | Neon and a background

Figure 3: After converting text strings to vector elements, you can shape them according to your imagination.

with that, use *Extensions | Modify Path | Jitter Nodes* (Figure 3, third diagram).

First, you need to convert the object deformed with the *Path Effects Editor* tool to a static path again with *Path | Object to Path*. This action ends any "live" changes to the envelope curves, but you can first modify them exclusively with further live effects, although not with common extensions.

Jitter Nodes works only if you previously used *Extensions | Modify Path | Add Nodes* to add nodes to the object at regular intervals. For the slight irregularities seen in the diagram, set *max. segment length* to *15.0* pixel separation. *Jitter Nodes* with a *Maximum shift* of *1* in both directions. You can activate *Shift nodes* and *Shift node handles* to get the handmade effect in Figure 3, third diagram.

Blazing Words

If you add a *Filter Effect* to the text (in the example, *Bevels | Neon*), the text is modified further (Figure 3, bottom diagram). You can then add a fiery bitmap background (Figure 5) in the next step.

Here is where Inkscape's powerful masking function comes into play (*Object | Mask | Set*). Whereas an edited path simply trims the background image, a mask brings the background image forward within the contours of the superimposed object bases the opacity on the lightness or darkness of the mask.

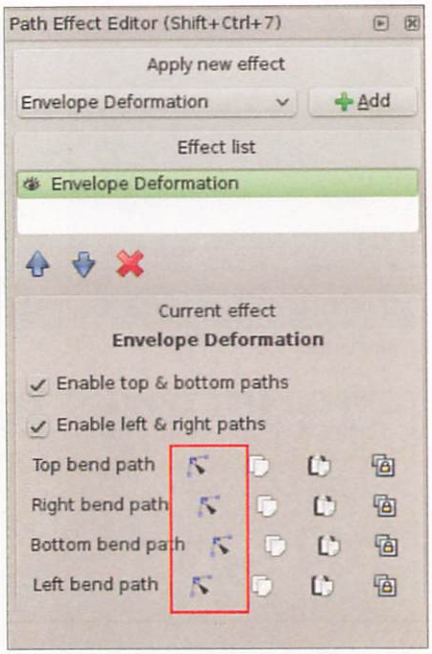


Figure 4: After clicking one of the four red nodes, you get an envelope curve path that can be formed like the usual vector path.

Dark parts of the mask yield a total transparency, while white parts remain opaque. Thus, you see the textured effect that the *Dark glass* filter adds to the *Ubuntu User* text in Figure 5. To do this, first import a fiery red bitmap into the Inkscape document. This image serves as the text background. Place the text over the background, select both, and use *Object | Mask | Set*. To have the text clearly visible (because the mask renders it semitransparent), you can also back it up with an effectless wine-red copy of the text.

Non-Identical Clone

The last example is based on the Inkscape tool *Edit | Clone | Create Tiled Clones*. By default, this tool generates a palette of simple evenly spaced objects (Figure 6, top).

However, you can vary many Inkscape parameters as desired, such as shift, color, scale, rotation, blur, and sharpness progressively in x and y directions (Figure 6, second diagram). The changes can be linear or expo-

Figure 5: A transparency mask applies the fiery background to the text.

nential and even random (Figure 6, third diagram).

You can use the clone function to rotate and mirror the objects with every cloning step (Figure 6, bottom). As Wikipedia clearly explains [5], the tool offers 17 possible ways to create repeating patterns through transformation, rotation, and mirroring. Inkscape uses all these symmetry methods, and they can accessed from the *Symmetry* tab in the toolbox dialog.

Cloning even works for backgrounds. Figure 7 shows a grid effect based on the brightness of a stored bitmap. Horizontal and vertical color and saturation changes in the small squares add a little color to the black-and-white photo.

The random generator rotates the objects at a maximum 10 percent and varies their intervals up to 20 percent so that the distribution is less uniform and looks more handmade. The opacity and size of the squares are determined by the brightness in the photo.

Point by Point

The speckled photo measures 725x875 pixels. A square of 7x7 pixels at the upper left of the photo serves as the basis for the grid in Figure 7. The clone tool can change the color only if both the color and fill are *unset* by choosing the question mark on the *Object | Fill and Stroke* dialog (Figure 8).

Next, open the *Edit | Clone | Create Tiled Clones* dialog while the little square is still active. On the *Symmetry* tab of the *Create Tiled Clones* tool, choose *Simple translation* without any rotation or mirroring.

To set the correct density for the grid, on the *Shift* tab, set the *Shift X* value to 100 percent per column; then, set the same value for the *Shift Y*, plus a *Randomize* factor of 20 percent in both cases (see Figure 9).

When you set the *Rows, columns* at the bottom of the dialog to 65 and 55, respectively, and then click *Create*, a slightly irregular pattern emerges that completely covers the bitmap.

To add a bit more variety, on the *Rotation* tab, you can enter a value of 100 percent for the *Angle* and 10 percent for *Randomize*.

On the *Scale* tab, then choose *Scale X* and *Scale Y* values of *-0.3* per row and column so that the splash of color shrinks slightly toward the lower right margin.

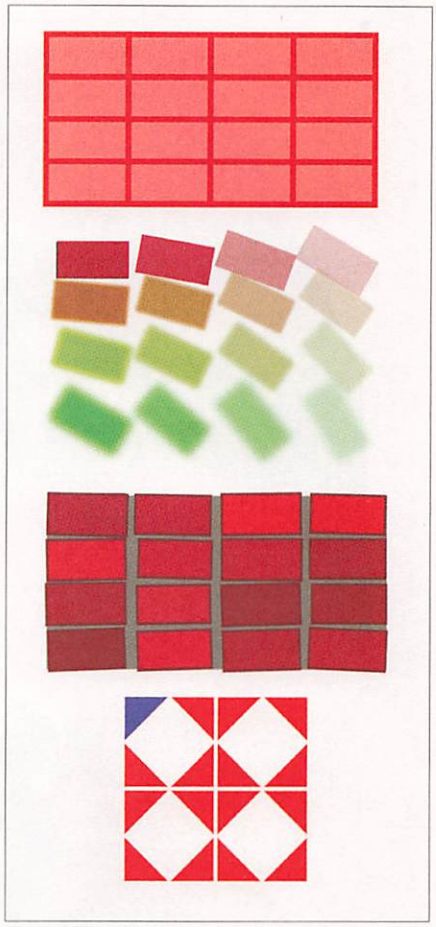


Figure 6: The tiled clone tool creates multiple objects and varies many clone properties.

Color Splash

On the *Color* tab, click the little field next to *Initial color* and add a turquoise shade. In Figure 7, you see that this step varies the base color, with the *H* ("hue") set to 1 percent per row and -1 percent for column, plus a randomize factor of 10 percent.

The *S* ("saturation") is reduced by -1 percent for every row and column in the example. When you click *Create* at this point, you will get a colored grid that is placed over the bitmap, similar to what you can see in Figure 7. However, the size of the grid points still doesn't match the brightness of the photo.

To do this, go to the *Trace* tab and enable the *Trace the drawing under the titles* option. With the *1. Pick from the drawing* option enabled, leave this as *Color* to set it as the criteria for the size of the grid points. On a black-and-white image, the *L* ("lu-

Figure 7: Because the tiled clone tool responds to the brightness curve of the bitmap, you can create many visually interesting grid effects.

minance") setting would have the same effect as described previously.

Under *2. Tweak the picked value*, using a *Gamma-correct* value of 3 would create a well-scaled brightness effect on the grid. Under *3. Apply the value to the clones*, the *Size* and *Opacity* selections provide a realistic grid effect.

The tiled clone tool can also theoretically be used for four-color printing grids. This example, which varies the image brightness by position and uses random values, still doesn't ensure correct color reproduction.

Additionally, the turquoise and purple tones enrich the black-and-white image with a few selected splashes of color, which is at least pleasing to the eye.

More to Come

In this article, I described several Inkscape features, such as filters, transparent masks, envelope curve deformation, and the jitter node extension along with the tiled clone tool. These are definitely important Inkscape functions that have creative potential, but I have really just scratched the surface of this powerful application.

Another definite advantage of the Inkscape SVG editor is that all three examples presented here, after being saved in *normal SVG* format, can be opened in modern web browsers without compromising their resolutions or the effects. •

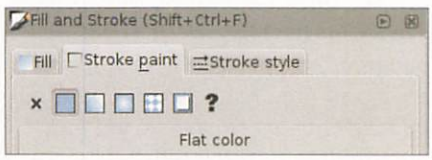


Figure 8: To make the tiled clone tool change colors, the base object needs to have a fill and stroke of unset.

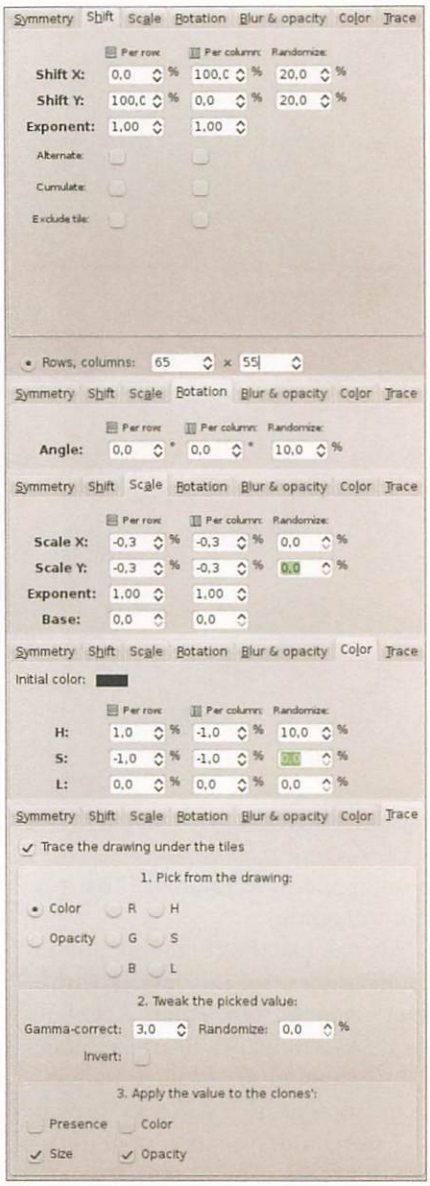

Symmetry Shift Scale Rotation Blur & opacity Color Trace
Per row: Per column: Randomize:
Shift X: 0,0 % 100,0 % 20,0 %
Shift Y: 100,0 % 0,0 % 20,0 %
Exponent: 1,00 1,00
Alternate:
Cumulate:
Exclude tile:
Rows, columns: 65 × 55
Angle: 0,0 ° 0,0 ° 10,0 %
Scale X: -0,3 % -0,3 % 0,0 %
Scale Y: -0,3 % -0,3 % 0,0 %
Base: 0,0 0,0
Initial color:
H: 1,0 % -1,0 % 10,0 %
S: -1,0 % -1,0 % 0,0 %
L: 0,0 % 0,0 % 0,0 %
Trace the drawing under the tiles
1. Pick from the drawing:
Color R H
Opacity G S
B L
2. Tweak the picked value:
Gamma-correct: 3,0 Randomize: 0,0 %
Invert:
3. Apply the value to the clones':
Presence Color
Size Opacity

Figure 9: The settings on the Create Tiled Clones dialog create the colored grid in Figure 7 based on the little square in the top left corner of the photo.

Info

[1] Inkscape handbook: *http://tavmjong.free.fr/INKSCAPE/MANUAL/html/index.php*

[2] "Inking It In" by Peter Kreussel, *Ubuntu User*, Issue 20, pg. 58.

[3] Persian wall mosaic: *http://www.sxc.hu/photo/465513*

[4] Tracing bitmaps: *http://tavmjong.free.fr/INKSCAPE/MANUAL/html/Trace.html*

[5] Symmetry schemes: *http://en.wikipedia.org/wiki/Wallpaper_group*

Discovery Guide

Look inside for a practical introduction to Ubuntu, including tips on printing and installing software, as well as installing, updating, and networking your Ubuntu system.

Installing Ubuntu 14.04 80

Network and Updates 84

Package Management 86

Multimedia 89

Installing Ubuntu 14.04

INSTALLATION PARTY

BY KRISTIAN KISSLING

Ubuntu 14.04 offers some changes to the installation process. We'll walk you through upgrading or installing the latest release.

If you have installed Ubuntu in the past, you might notice that, unlike its predecessors up to version 12.04, Ubuntu 14.04 does not fit on one CD.

As from 12.10 onwards, developers also improved the installer, which now provides full encryption and Logical Volume Manager (LVM). The step in 13.10 that allowed access to or set up of an UbuntuOne account has been removed. Other than these updates, not much has changed with the Ubuntu installation process.

Installing Ubuntu 14.04 on your computer or virtual machine can be done in several ways. Although Ubuntu is geared toward novice Linux users, installing the system is different than a Windows installation. In this article, we'll explain your options and walk you through the steps.

What's New, Trusty?

If you're running Ubuntu 13.10, you can update to version 14.04 directly. However, if you're upgrading from a version of Ubuntu earlier than 13.10, you will have to update to 13.10 first and then to 14.04.

If you decide to go with a clean install, try the Live version first to test the programs and get used to the Unity desktop. Then you can load the distribution via the integrated installer on your computer.

Regardless of what you decide, you should back up all your important files first so that nothing gets lost in the process.

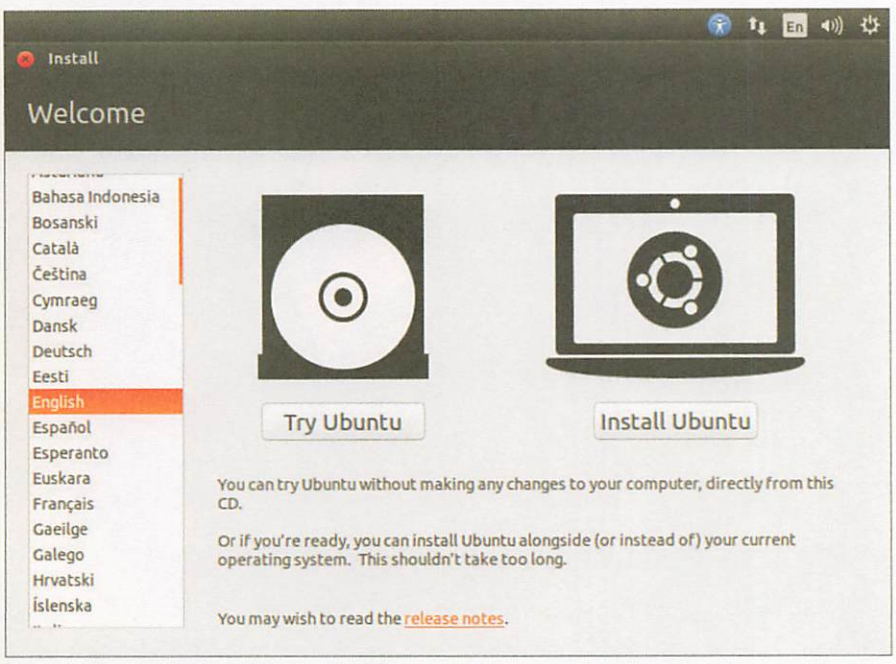


Figure 1: Choose "Try Ubuntu" to run a Live instance on your computer without installing anything.

Insert Media

The double-sided DVD that comes with this issue provides two versions of Ubuntu, for 32-bit and 64-bit computers. To boot from the DVD, first have your computer recognize it as bootable. Often computers do this automatically; starting the PC with a bootable CD or DVD in the drive invokes the corresponding bootloader.

If the computer cannot find the DVD, you can change the boot order via the BIOS. To get to the BIOS, start up the computer and immediately press Esc, Del, F2, or F12, depending on your computer. Then look for the appropriate option to change the boot order and move the CD/DVD drive to the top of the list.

If you're starting Ubuntu from a USB stick, move the USB device to the top of the boot order. In both cases, save the setting and restart the computer. If your computer is missing a drive, use a USB stick.

We will assume that you already have Ubuntu installed. Create the stick by downloading the ISO image for Ubuntu 14.04 for the 32-bit or 64-bit architecture [1] that your installed Ubuntu uses, then load it to the USB stick using the *Startup Disk Creator* app in Ubuntu (Figure 2).

To begin, delete all previous files on the stick, leaving at least 4GB of free space. At the top of the *Make Startup Disk* window select the downloaded ISO image and at the bottom select the USB device to use, then click *Make Startup Disk*. Booting from the USB

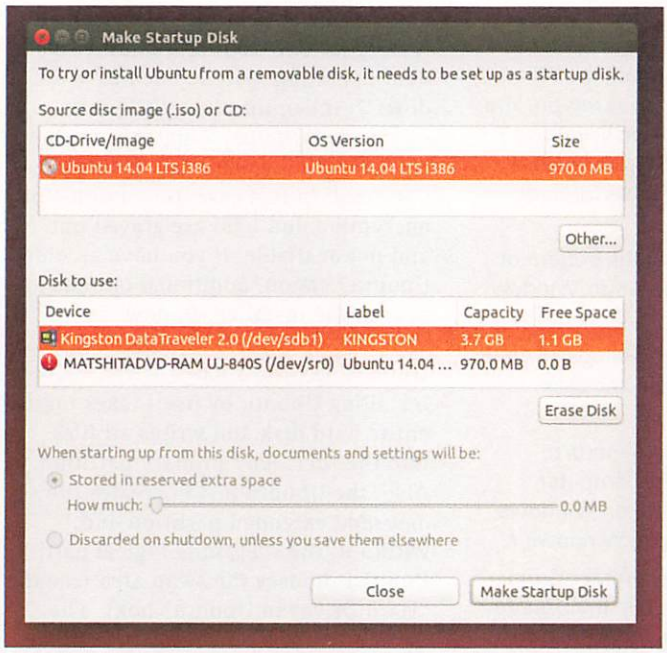


Figure 2: The Startup Disk Creator in Ubuntu 13.04 copies the Ubuntu 13.10 image onto a USB stick with the required free space. Both Ubuntu versions must use the same 32-bit or 64-bit architecture.

stick requires the BIOS setting previously mentioned.

Fresh Install

To do a new installation, insert the DVD included in this issue into the drive with the DVD label showing the appropriate architecture for your system (either *32-bit* or *64-bit*), and then turn off the computer.

Turn it back on and immediately hold down the Shift key and you will get to the Ubuntu boot menu. Here you can make a language selection using the *F* keys to set language, keymap, and boot parameters.

Press F6 (*Other Options*) on the Ubuntu boot menu so that you do not land on the Ubuntu desktop. The window has a *Boot Options* line at the bottom (Figure 3).

Enter the boot options mentioned here instead of the default ones *quiet splash*. The *nomodest*, *radeon.modeset = 0* or *nouveau.modeset = 0* options help if you have graphics problems and the screen stays black. The *acpi = off* boot option fixes power management problems on older computers.

Ubuntu on Trial

If you do not want to *Install Ubuntu* right away, you can *Try Ubuntu without installing*. In so-called Live mode, Ubuntu copies only the necessary programs into memory and executes them from there.

Booting takes noticeably longer than the installed version, and the system as a whole is a bit slower, but you can test Ubuntu 14.04 with little risk. Also look at the bundled applications to see whether Ubuntu supports your hardware (printers, scanners, wireless, and graphics cards).

Ubuntu 14.04 loads only the 3D version of the Unity desktop and uses LLVMpipe for slower machines, creating sluggish responses on some older computers (cause for a recommended desktop replacement).

Hard Drives in Ubuntu

Ubuntu names device files on hard drives according to a particular schema: */dev/sda* addresses the first detected hard drive, with */dev/sda1* its primary partition (*/dev/sda2* through */dev/sda4* are the three other primary partitions). The */dev/sda5* device is the first logical partition located in the (*/dev/sda2* extended partition. Windows should always be located in the first primary partition (*/dev/sda1*), and generally also in */dev/sda2*. If you plug a USB stick or external hard drive in the USB slot, these external storage devices get the designation */dev/sdb*, with everything else following the same schema. Using the *sudo fdisk -l* command, you can get an overview of the available hard disk and USB memory and the partitions and filesystems within them.

Installer

The Ubuntu installer opens in a window as shown in Figure 4. First choose a *language*. The next window checks to see whether your computer meets the installation requirements, which is at least 5.9GB available drive space, plugged in (important for laptop and netbook installations), and preferably connected to the Internet.

In this window, you can set two checkmarks. If you mark *Download updates while installing*, Ubuntu uses the idle time during installation to download patches and security up-

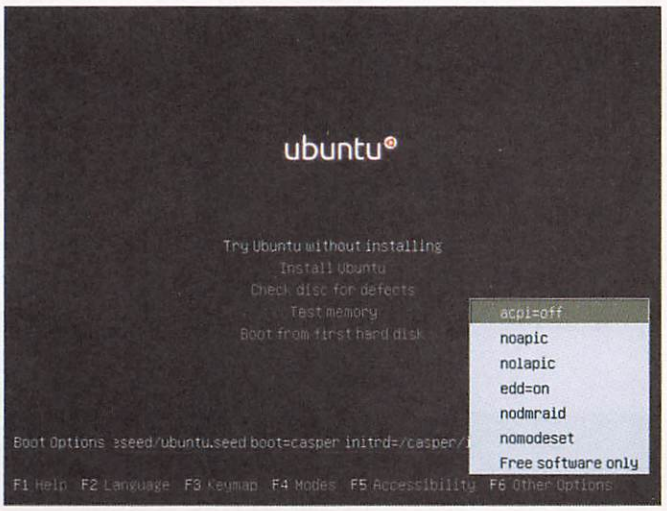


Figure 3: If booting doesn't work, adjust the boot options in the Ubuntu boot menu.

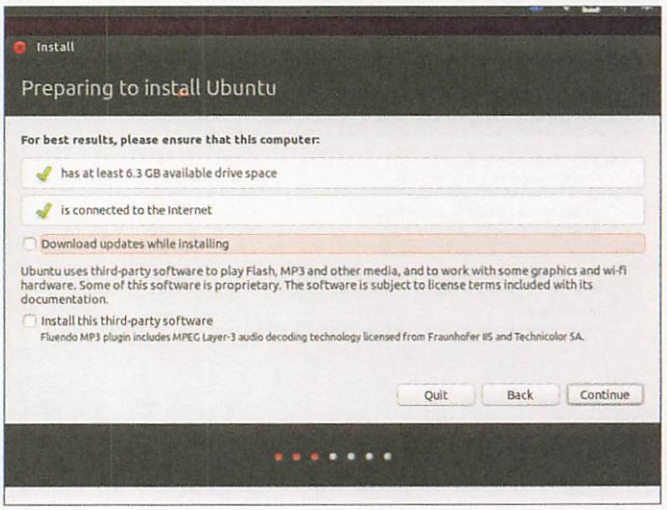

Install
Preparing to install Ubuntu
For best results, please ensure that this computer:
has at least 6.3 GB available drive space
is connected to the Internet
Download updates while installing
Ubuntu uses third-party software to play Flash, MP3 and other media, and to work with some graphics and wi-fi hardware. Some of this software is proprietary. The software is subject to license terms included with its documentation.
Install this third-party software
Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Technicolor SA.
Quit
Back
Continue

Figure 4: The installer provides options, among other things, to prepare for MP3 support and for downloading updates.

Windows and Ubuntu

Windows likes to save its files across the entire hard disk. By defragging the Windows partition, you arrange the scattered file fragments in a contiguous region. You can then shrink the partition without disrupt any installed programs. In Windows 7, you can start the defragging with *Start* | *Programs* | *Accessories* | *System Tools*. If you freshly installed Windows, a program called *chkdisk* routinely looks for damages to the partition, and then Windows can boot normally.

After defragging, you can begin shrinking the Windows partition. At the bottom of the Start menu, type *hard* and click *Create and format hard drive partition*. Window XP and Vista have similar functions.

Next select the Windows partition you want to shrink. Right-click and choose *Shrink Volume*, enter a desired value in the *Total size after shrink in MB* field and click *Shrink*, and restart Windows.

In any case, after you install Ubuntu and restart Windows, Windows resorts to *chkdisk* to check for possible damages. The system then restarts the computer again before starting up normally. Wubi [2], the tool to install Ubuntu on Windows, is included with Ubuntu 14.04 LTS, and you will be able to use it up to Windows 7.

Installing Windows after Ubuntu is so complicated that we don't recommend you do it. But, if you do plan to do so, choose creating a manual partition in the Ubuntu installer (as described in the "DIY Ubuntu" box) while reserving an empty */dev/sda1* partition for Windows.

dates from the Internet, but doesn't install them yet.

Marking *Install this third-party software* loads during Ubuntu installation a few important codecs and plugins for your system. These codecs and plugins will allow you to play MP3 files on your player and see flash content in your browser. These two options can always be set later, so, for now, *Next* takes you to the next window.

If a wired connection exists to the network and Ubuntu recognizes your wireless card, the next window displays the discovered access points. Using the correct password, you can connect to the access point. If no choices are available, a wireless hardware is possible, so use a wired connection instead.

Choices, Choices

A new window then appears, which was introduced in Ubuntu 12.10. If Ubuntu doesn't find another operating system installed, it provides the options, as before, to *Erase disk and install Ubuntu* or to do *Something else* (Figure 5).

As from 12.10, there are two additional options: *Encrypt the new Ubuntu installation for security* and *Use LVM with the new Ubuntu installation*. If you encrypt Ubuntu, the performance is a bit less than optimal, although it secures data against theft. Note that you can always encrypt your home directory later. If you have another resident operating system, such as Windows 7, the option *Install Ubuntu alongside Windows 7* appears on the screen, which you can invoke using *Continue*. In this case, the options for encryption and LVM are grayed out and not available. If you have an older Ubuntu version, additional options appear in the installer window.

Ubuntu Flying Solo

Installing Ubuntu by itself takes up the entire hard disk and writes all files into the */dev/sda1* primary partition. Also, the Ubuntu install creates the */dev/sda2* extended partition and, within it, the */dev/sda5* logical partition that houses the swap area (see the "Hard Drives in Ubuntu" box). The swap area includes the swap file that takes over when working memory is overloaded, although writing to the swap area is noticeably slower than direct access to main memory.

Ubuntu Neighbors

Windows often takes up the entire hard drive instead. If you want to install Ubuntu alongside Windows, you must shrink the Windows partition and completely defrag it, as described in the "Windows and Ubuntu" box. Ubuntu 14.04 then invokes the GRUB 2 bootloader, which boots both systems and can handle new computers supporting Secure Boot.

DIY Ubuntu

If you want to maintain some control over the installation, select the *Something else* option, although this re-

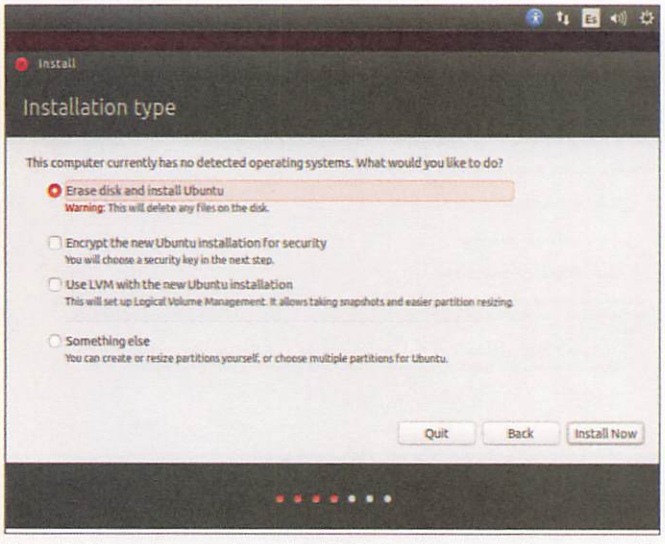

Install
Installation type
This computer currently has no detected operating systems. What would you like to do?
Erase disk and install Ubuntu
Warning: This will delete any files on the disk.
Encrypt the new Ubuntu installation for security
You will choose a security key in the next step.
Use LVM with the new Ubuntu installation
This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.
Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.
Quit
Back
Install Now

Figure 5: Choose to erase the entire disk for the Ubuntu install or make a different selection.

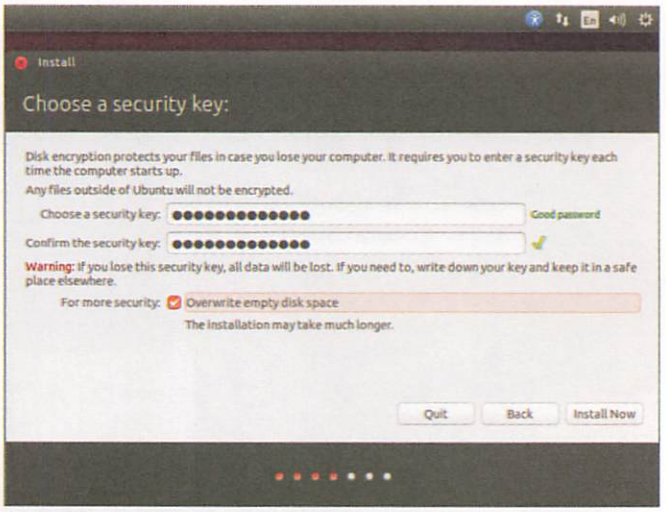

Install
Choose a security key:
Disk encryption protects your files in case you lose your computer. It requires you to enter a security key each time the computer starts up.
Any files outside of Ubuntu will not be encrypted.
Choose a security key:
Good password
Confirm the security key:
Warning: If you lose this security key, all data will be lost. If you need to, write down your key and keep it in a safe place elsewhere.
For more security:
Overwrite empty disk space
The installation may take much longer.
Quit
Back
Install Now

Figure 6: If you choose the complete encryption option that Ubuntu offers, be sure to use a secure password (security key) and keep it in a safe place so as not to lose it.

quires some knowledge of the Ubuntu hard drive structure (see the "Hard Drives in Ubuntu" box). Ideally you would place at least two logical partitions for Ubuntu on an empty hard drive: a swap partition (*/dev/sda5*) and a partition for the *root* directory that is at least 4.9GB (*/dev/sda6*).

Optionally, add an extra partition for the */home* directory (*/dev/sda7*) that gives you enough space for all your files. This partitioning has the advantage that you can install Ubuntu again later without losing your user data. Add the new partitions using *Add*. For the / and */home* directories, selecting the *Ext4* filesystem is best.

Install Now

If you opted for self-partitioning, click *Install Now*. To set up complete encryption, Ubuntu asks for a security key. Select the most secure one possible (Figure 6).

Ubuntu then asks about your location (*Where are you?*, to get the time zone), *Keyboard layout*, name, username, and password (*Who are you?*). The latter dialog also asks whether you want to *Require my password to log in* or *Log in automatically*, with an additional option to *Encrypt my home folder*. Ubuntu uses EcryptFS for this.

The following screens are for selecting your timezone, keyboard and setting your username and password. If you have one of the supported Ubuntu webcams, the installer also suggests that you take an image of yourself to complement your profile. Ubuntu 14.04 can also import profiles from other installed operating systems, including not only browser and email files, but also screen backgrounds and documents.

During the install process, you can view a slideshow that presents the new features in Ubuntu. Installation can take a while, depending on your system, and it ends with a prompt to *Restart Now*. Remove the CD/DVD and press Enter so that the computer reboots. If you activated system encryption, enter the password before booting up.

Updating Ubuntu

If you have Ubuntu 12.04 installed, first consider whether you want to upgrade at all. Version 12.04 is a LTS version, so it will still be supported until 2017. The new Ubuntu 14.04 is also a LTS version with five years of support. If you still want to upgrade to 14.40, you may have to upgrade to the intermediate versions first.

Before beginning, turning off the external repositories and PPAs first makes sense. Use the Software Center *Edit | Software Sources* function and remove the check marks on the *Other Software* tab.

Next, press Alt + F2 and enter *update-manager -d*. In the empty window of the update manager, an *Upgrade* button shows at the top. During the upgrade, the installer asks whether you want to keep or replace a specific configuration file; choose to replace if you haven't made any changes. Because Ubuntu is downloading many packages and replacing them, make sure your computer is plugged in to stay charged.

Bootloader Failures

Occasionally bootloader doesn't appear after an Ubuntu installation, or you can boot only one system. Loading Windows *after* installing Ubuntu often leads to a bootloader failure. Refer to the "Revival" box for commands you will want to enter in a terminal window. These commands install the GRUB 2 bootloader with help from a Live system so that GRUB correctly identifies all systems. •

Info

[1] Download Ubuntu 14.04: *http://releases.ubuntu.com/trusty/*

[2] WubiGuide: *https://wiki.ubuntu.com/WubiGuide*

Revival

To revive an installed Ubuntu, boot up Live mode on the DVD in this issue by choosing the *Try Ubuntu without installing* option. Be sure that the Live and installed versions are the same architecture. Then invoke a terminal and execute the following commands. Substitute for */dev/sda6* entry the specific partition in which the root directory (*/*) of the installed Ubuntu is located.

```
$ sudo mount /dev/sda6 /mnt
$ sudo mount -o bind /dev /mnt/dev
$ sudo mount -o bind /sys /mnt/sys
$ sudo mount -t proc /proc /mnt/proc
$ sudo chroot /mnt
$ grub-mkconfig -o /boot/grub/grub.cfg
$ update-grub2
$ grub-install /dev/sda
$ exit
$ sudo reboot
```

Use the *chroot* command (change root) to mount the installed system into your Live system and work with it as if you had booted it. Now, you can set up a new GRUB 2.

DIY Ubuntu

If you want to maintain some control over the installation, select the *Something else* option, although this requires some knowledge of the Ubuntu hard drive structure (see the "Hard Drives in Ubuntu" box). Ideally, you would place at least two logical partitions for Ubuntu on an empty hard drive: a swap partition (*/dev/sda5*) and a partition for the *root* directory that is at least 4.9GB (*/dev/sda6*). Optionally, you can add an extra partition for the */home* directory (*/dev/sda7*) that gives you enough space for all your files. This partitioning has the advantage that you can install Ubuntu again later without losing your user data. Add the new partitions using *Add*. For the / and */home* directories, selecting the *Ext4* filesystem is best.

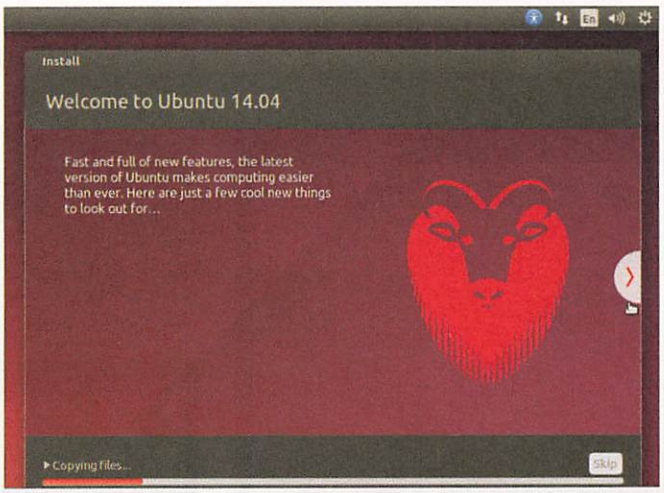
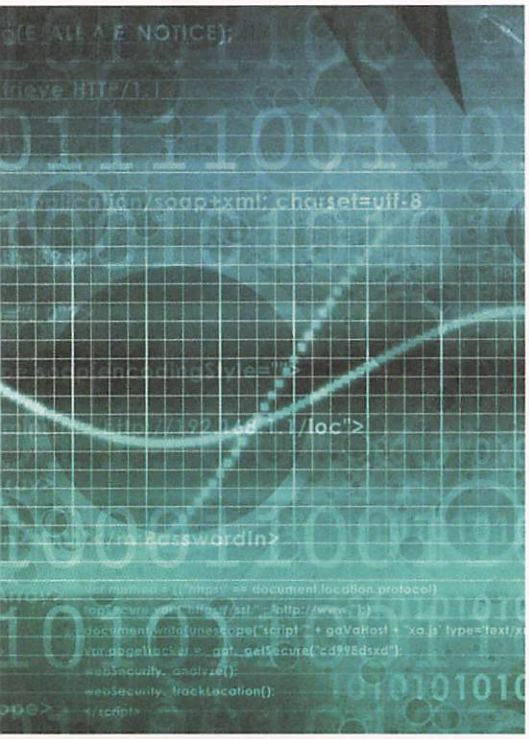

Install
Welcome to Ubuntu 14.04
Fast and full of new features, the latest version of Ubuntu makes computing easier than ever. Here are just a few cool new things to look out for...
Copying files...
Skip

Figure 7: New in 13.10 is the option to configure your Ubuntu One account during the install.



NetworkManager and system updates

Active Traffic

BY KRISTIAN KISSLING

With NetworkManager, you can access the Internet during the Ubuntu installation process. The first step is to download updates to make your system more secure.

After the official release of a distribution, the developers continue to iron out bugs with various patches, and they continue to minimize security vulnerabilities during the entire support period of the operating system. Ubuntu 14.04, for example, is a LTS release and will be supported for the next 5 years. If you install the distribution a few weeks after its official release, updates are already available.

To look for updates, call up the *Software Updater* program (Figure 1), which displays a list of installable patches. Clicking *Install Now* downloads the patches onto your computer (Figure 2). Restart the system to apply the patches. If no updates are available, you will see *The software on this computer is up to date* (Figure 3).

Bug fixes help keep your system safe and secure. With a few exceptions, such as Firefox and Thunderbird, newer versions of installed programs are updated with the next Ubuntu version. Within the support cycle of Ubuntu 14.04, only Backports and PPAs [1] can be applied to a new version. Backports are special repositories that provide newer versions of some software [2].

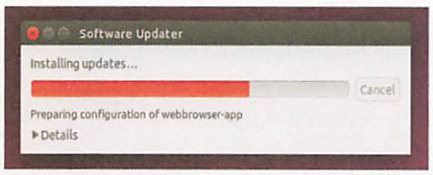


Figure 1: If Ubuntu updates are out there, you'll find out by invoking the "Software Updater" program.

Figure 2: Ubuntu shows details on what updates are waiting in the queue.

Chicken and Egg

To get updates for Ubuntu 14.04, you must have access to the Internet. The *NetworkManager* helps you set up that access during the installation. If you're not cabled in, the installer presents a window listing the neighboring wireless networks and asks you to connect to one, which is when installation begins in earnest.

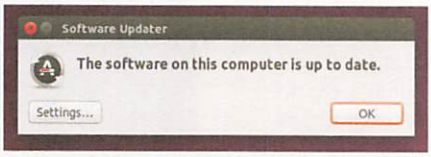


Figure 3: No updates are needed and you are good to go.

To bypass this step, install Ubuntu without Internet access, and then later set it up through the NetworkManager, which is represented by a small icon at the top right of the screen. If you're using a wireless card, you will see a wave-shaped icon and, if the system is connected, the icon reflects the quality of the connection. Since Ubuntu 12.10, connecting to the Internet even before signing onto the desktop is possible.

Wireless Problems

Not all wireless cards work flawlessly with Ubuntu. Before you install Ubuntu on your computer, use the Live version to test whether Ubuntu recognizes your wireless chip. Boot into Live mode and try to connect to one of the wireless networks displayed. If connecting fails or no access points are showing, check the hardware switch on the computer itself.

If you run into problems configuring wireless cards, DSL modems, or UMTS sticks because NetworkManager doesn't recognize them, temporarily connecting the computer with a network cable to a switch or router usually works.

Hardware is a frequent source of wireless problems. On many notebooks, a button turns on the wireless card, whereas on others, a special key combination (such as Fn + F5) or a single *F* keystroke works. Test to see whether the WLAN has been deactivated by mistake.

To enable access for DSL, UMTS, and so on, click in the upper right on the NetworkManager icon and choose *Edit connections*. Through *Connection information*, you can find out whether the access point provided you an address via DHCP (Figure 4). DHCP (Dynamic Host Configuration Protocol) is the service that automatically assigns you

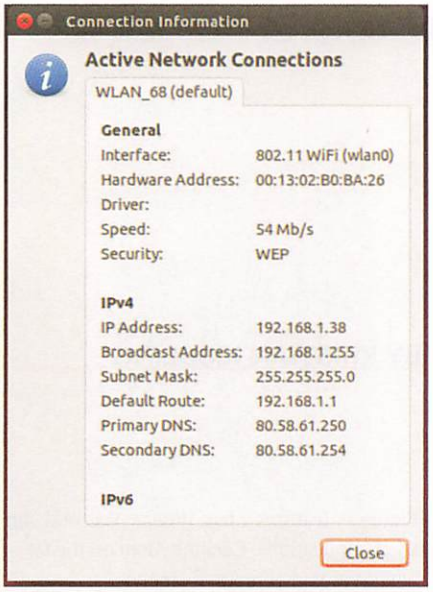


Figure 4: Connection Information provides some helpful connectivity facts.

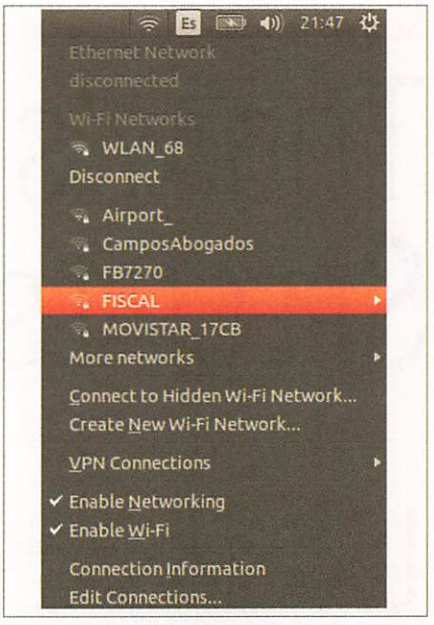


Figure 5: NetworkManager displays the neighboring access points, which is generally the first choice when accessing the Internet..

an IP address in the local net. Most routers provide DHCP addresses by default.

Ethernet

Wireless routers usually have inputs for network cables. If the router assigns IP addresses with DHCP, which it usually does, NetworkManager automatically connects after you plug in the network cable. Rarely does this not work, but if that is the case, you must assign an IP address and domain name manually.

Wireless Networks

So that Ubuntu can set up your wireless card properly, the kernel needs to find a suitable driver for the card. Then it's as easy as clicking the NetworkManager to obtain the neighboring networks (Figure 5). Select the name of the wireless network and log in with your passcode.

To avoid mistyping the password, make the password visible. Ubuntu automatically detects the encryption type (WEP or WPA) the access point uses. After a successful network authentication, your wireless card provides you a valid IP address via DHCP.

Setting up your smartphone (Android or iPhone) as an access point, which is called *tethering*, also works. NetworkManager treats the tethered smartphone as a normal access point and connects to it. Smartphones use the Universal Mobile Telecommunications System (UMTS) to get to the Internet, which is a good solution when you are on the move. Remember, however, to secure the access point on your mobile phone with a good password and deactivate it when you are done.

If the kernel doesn't provide a driver for your wireless card, a proprietary driver is usually available. To test this, press the Windows key and open the *Software Repositories* program. If the *Additional Drivers* tab lists the driver for your card, click *Activate* so that the wizard can download the corresponding software for the card.

If both steps don't work, search online for *Ubuntu 14.04 {YOUR WIRELESS CHIP}*. To find out what your built-in wireless chip is called, press Ctrl + Alt + T to open a terminal and enter:

```
$ lspci | grep "Network"
```

For a USB device, use the following command to find the chip's *ID* number:

```
$ lsusb
```

An Ubuntu help page [3] lists the wireless card and its degree of support in Ubuntu and its derivatives. Setting up an unsupported wireless cards can be complicated on occasions, and other times a simple shell command can activate a card. In some situations, programs such as *Ndiswrapper* allow use of the card's Windows driver with Ubuntu [4].

UMTS Cards and DSL Modems

Using NetworkManager, you can also set up UMTS cards and chips, as well as DSL modems. Of course, the system must recognize the hardware, otherwise the tabs are grayed out. Once Ubuntu identifies a UMTS card, insert the card into the USB slot, enter the PIN, and configure it using NetworkManager.

DSL modems are also candidates for NetworkManager. Click *Edit Connections* and choose *Add* on the *DSL* tab, connect the DSL modem to the phone jack and your computer via cable to the modem, and apply the access credentials from your Internet service provider. These will go on the *DSL* dialog.

Usually you will find a long line of text among the data, including the @ character. The part before the @ is your *username*, and the part after it is the *service name*. Then enter the passcode and change to the *Wired* tab, where you enter the MAC address of your network card connected to your DSL modem. The name is usually *eth0*. After you determine the address, open a terminal and enter:

```
$ /sbin/ifconfig | grep Hardware
```

The MAC address consists of six two-character sections separated by colons. Enter the MAC address in the field for the DSL configuration.

VPN Connections

If you want to surf securely on the Internet through a tunnel, use a virtual private network (VPN) solution such as OpenVPN, which NetworkManager also supports. Details on OpenVPN and how it works with Ubuntu are in an article in *Ubuntu User* issue 13, page 58-59.

Troubleshooting

If the DSL or UMTS modem still gives you problems despite the correct driver, NetworkManager may have a bug. One solution is to install a recent version of the software by integrating a PPA. This package often provides a newer version than the one normally used by Ubuntu. Often you can solve network problems by importing all updates for your release and restarting the computer. •

Info

[1] Packaging/PPA: *https://help.launchpad.net/Packaging/PPA*

[2] UbuntuBackports: *https://help.ubuntu.com/community/UbuntuBackports*

[3] WifiDocs/WirelessCardsSupported: *https://help.ubuntu.com/community/WifiDocs/WirelessCardsSupported*

[4] WifiDocs/Driver/Ndiswrapper: *https://help.ubuntu.com/community/WifiDocs/Driver/Ndiswrapper*

[5] Using OpenVPN Under Ubuntu: *http://www.ubuntu-user.com/Magazine/Archive/2012/13/Using-OpenVPN-Under-Ubuntu*

Software management on Ubuntu 14.04

TODAY'S SPECIALS

BY KRISTIAN KISSLING

Ubuntu's Software Center lets you install just about anything, including free software, purchased apps, and games.

Ubuntu includes the convenient Software Center, which provides centralized software management and allows you to search for new software and install it with a mouse click. With an Ubuntu One account, you can purchase software with a credit card.

The Debian packages (that end in *.deb*) that Software Center offers are actually archive files containing libraries, configuration files, and executable programs. If you install the Firefox package, for example, the package manager distributes the necessary files to the correct places in the filesystem.

Often there are dependencies, which means that installing program *A* requires that you also have program *B* installed. The package manager identifies the dependent packages automatically during installation and loads them on the computer. Usually you will notice the dependencies when you try to install a single Debian package over the Internet.

The package you are looking for, such as a video codec, may be missing in the Software Center. In this case, you can find the package in external repositories created by the software's users or developers. The repositories contain not only the software, but also the packages on which they depend. (How to tap into these repositories is covered in the "What are Repositories?" section.)

Figure 1: Software Center encourages you to browse.

PACKAGE MANAGER IN ACTION

Getting to repositories and installing software is often faster from the command line. The "Quick sources" section goes into greater detail. First, let's look at managing packages using Ubuntu's Software Center. To access the Software Center, press the Super key [1] and entering *softw*. The start page (Figure 1) features a few *What's New* and *Top Rated* applications. Clicking *More* on the far right gives you additional choices.

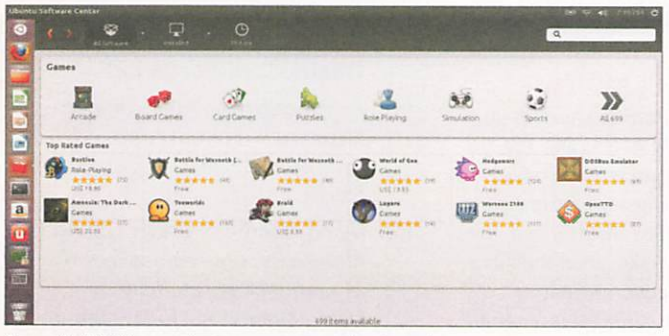


Figure 2: Clicking a category opens up subcategories.

If you click *Turn On Recommendations* at the bottom, Ubuntu suggests opening a user account at Ubuntu One. Then your computer regularly sends lists of software you've installed to Canonical and the company returns a list of software that might interest you. These recommendations appear on the start page and can be deactivated with *View | Turn Off Recommendations* in the global menu. In the meantime, you can purchase programs using your Ubuntu One account.

Use the arrow icons in the upper left to return to the start window. Starting with the release of Ubuntu 12.10, Ubuntu developers have placed ads for free and commercial software from the Software Center across the broad surface of the screen under the heading *Our Favorites*.

The left side of the Software Center screen organizes the applications by category. Clicking a category opens subcategories (Figure 2). All other controls are at the top of the package management window. The *All Software* menu lists all the software in all the available repositories. If you click the little white triangle next to the menu item (Figure 3), you'll find the entries *Provided by Ubuntu* (mostly free software), *Canonical Partners* (free and commercial software without source code, such as Skype and Flash), and *For Purchase* (commercial software, including many games).

Clicking *Installed* on the menu bar at the top displays only the installed applications, sorted by category. Click the little triangle to the right to view the packages from a single source. At the far right is the *History* menu item that tells you which software you downloaded. Check here if an upgrade suddenly isn't working. Downgrading, unfortunately, doesn't work in Software Center. To do that you must install the Synaptic Package Manager and specify an earlier *Version* in the *Properties*.

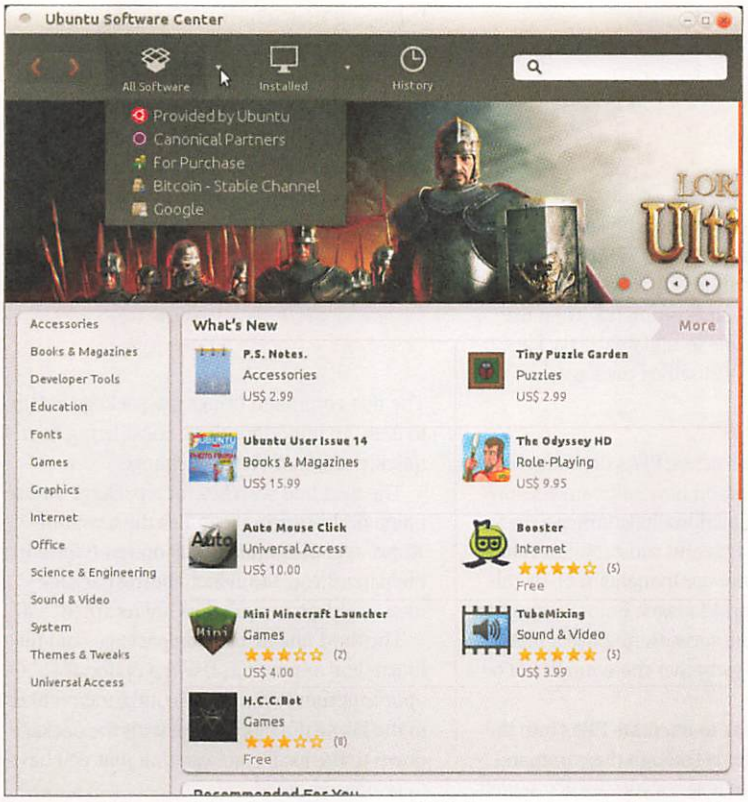


Figure 3: Clicking the little white arrow next to "All Software" shows the integrated repositories.

If you already know the name of the software, search for it in the search field at the upper right. Results come from the repository that you use for *All Applications*. Next to the software names are the ratings in the form of stars. Once you select a package, the *More Info* and *Install* buttons appear. The first button leads to a detailed description of the program, often showing a screenshot (Figure 4). Look under the *Add-ons* section to see software that could enhance the program. Here *People also installed* displays other programs to consider. You can also filter reviews by language is. Finally, click the *Install* button to install the program. Once installed, you can click the *Write your own review* link to evaluate the software. This requires a Launchpad account [1].

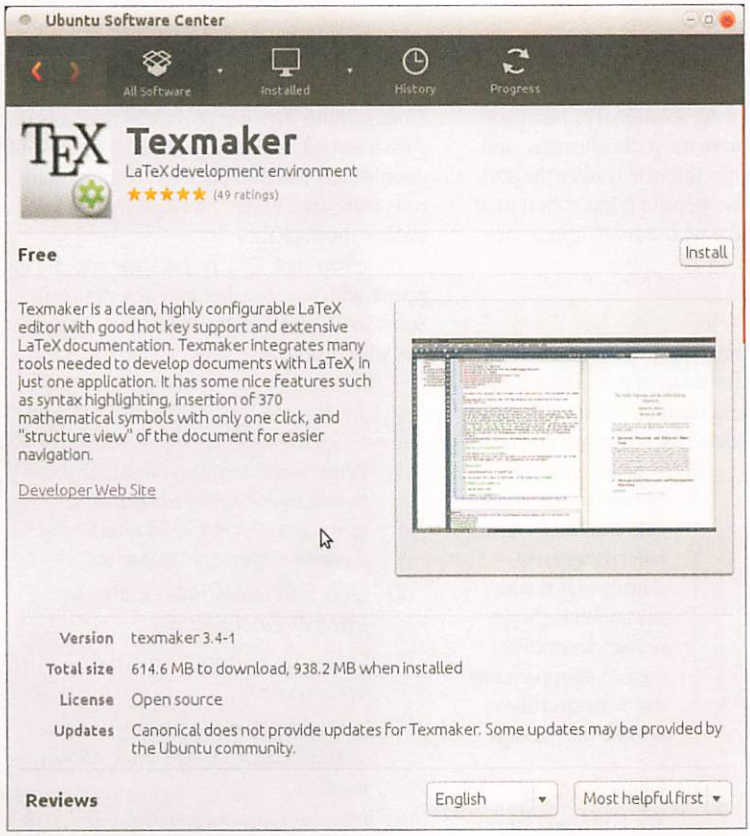

Ubuntu Software Center
All Software
Installed
History
Progress
Texmaker
LaTeX development environment
(49 ratings)
Free
Install
Texmaker is a clean, highly configurable LaTeX editor with good hot key support and extensive LaTeX documentation. Texmaker integrates many tools needed to develop documents with LaTeX, in just one application. It has some nice features such as syntax highlighting, insertion of 370 mathematical symbols with only one click, and "structure view" of the document for easier navigation.
Developer Web Site
Version texmaker 3.4-1
Total size 614.6 MB to download, 938.2 MB when installed
License Open source
Updates Canonical does not provide updates for Texmaker. Some updates may be provided by the Ubuntu community.
Reviews
English
Most helpful first

Figure 4: Once you choose an installable package, you'll receive further information: ratings, screenshots, descriptive text, and more.

The search won't find all programs. Software Center hides libraries, developer files, and even programs without graphical interfaces, such as the *mplayer* command interface. To install these programs, choose *Show technical items* at the bottom of the window or enter the full name of the software in the search field (Figure 5).

The Software Center global menu includes other interesting options. In the *File* menu, you can choose *Reinstall Previous Purchases* or *Sync Between Computers* to move your package selections to other computers over the Ubuntu One cloud service. Also, you can use *Edit | Software Sources* to add other repositories. If you don't want to display new applications automatically in the Launcher, unmark the *New Applications in Launcher* check box in the *View* menu.

What Are Repositories?

Software Center gives you access to thousands of packages, but the one you want might be missing or only in an older version. This is where external repositories and the PPAs help. The Opera repository [3] is one of the external repositories and provides the Opera web browser that is missing from Ubuntu's standard repositories. The VirtualBox project [4] also provides the proprietary version of its own software, which supports the virtual USB 2.0 ports and Remote Desktop Protocol (RDP), from its own repository.

Personal Package Archives (PPAs) are special versions of external repositories found only on Ubuntu's Launchpad. When experienced Ubuntu users or developers create Debian packages out of their favorite software, they frequently offer them in their personal package archive on Launchpad. Using a search function [5], you can peruse the platform for new versions of current software with great success.

Complementing Package Sources

In Software Center, you can download from external repositories using *Edit | Software Sources*. To use the Opera project's external sources, for example, click *Other Software* and *Add*, then enter *deb http://deb.opera.com/opera/ stable non-free*. This address wasn't invented: usually, the address is the website for the external repository. Click the *Add Source* button to accept the repository (Figure 6).

External repositories often protect themselves from abuse with keys. Every package has a signature verified with a key, so that manipulated packages with mismatched signatures fall by the wayside. Often you can't access a package offering without a key, but how do you find the key? To import the key

Figure 5: Additional libraries, developer files, and programs are listed as technical items.

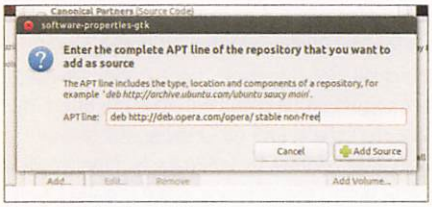

Canonical Partners (Source Code)
software-properties-gtk
Enter the complete APT line of the repository that you want to add as source
The APT line includes the type, location and components of a repository, for example 'deb http://archive.ubuntu.com/ubuntu saucy main'.
APT line:
deb http://deb.opera.com/opera/ stable non-free
Cancel
Add Source
Add...
Edit...
Remove
Add Volume...

Figure 6: Obtain additional packages from external repositories.

for the Opera repository, invoke a terminal through Ctrl + Alt + T and use the following:

```
$ sudo apt-get update
```

The command updates the list of available software for Software Center and returns an error message. Then, you can enter:

```
$ sudo wget -O -
  http://deb.opera.com/archive.key
  | sudo apt-key add -
```

For other repositories more tied in with Ubuntu, you may have to query the Ubuntu keyserver:

```
$ sudo apt-key adv
  --recv-keys --keyserver
  hkp://keyserver.ubuntu.com:80
  B9A06DE3
```

In this case, the above downloads the appropriate key (B9A06DE3) for the developer's Launchpad PPA for the Inkscape vector design software and integrates it with package management.

If you can't get to the keyserver, it may be temporarily overloaded. Replace *keyserver.ubuntu.com* in the command with another keyserver, such as *subkeys.pgp.net*, *pgp.mit.edu*, or *keys.gnupg.net*. After the *sudo apt-get update* command, you should be able to install the software from the source. The new repository might not show up in Software Center right away.

Key Over Terminal

As seen above, most websites show a simple line that you can copy with Ctrl + C and Ctrl + V to the terminal. The line might appear as in the following VirtualBox example:

```
$ wget -q http://download.
  virtualbox.org/virtualbox/
  debian/oracle_vbox.asc -O - |
  sudo apt-key add -
```

The command downloads the key from the website (using *wget -q {HTTP-LINK}*), sends it to the console (with *-O -*), and passes it to the *sudo apt-key add* command. Then use *sudo apt-get update* to update the package list and load up the VirtualBox package.

Integrating PPAs

With about 6,000 active PPAs on Launchpad, it is easy to get brand new software versions with the latest functionalities. Among them are exotic programs and (unstable) developer versions of system components such as the kernel or NetworkManager. Before compiling software from the sources, first use a PPA search [4] to see whether there might not be a PPA for it.

The fastest way to integrate PPAs into the package manager is through the command interface because it saves you the key search. Press Ctrl + Alt + T to open the terminal and enter what usually amounts to three command lines, something like the following:

```
$ sudo add-apt-repository
  ppa:inkscape.dev/stable
$ sudo apt-get update
$ sudo apt-get install inkscape
```

This happens to mount the PPA for the Inkscape vector drawing solution. For other software, simply replace the parts after *ppa:* and *sudo apt-get install*. Take the text for the part after *ppa:* from the website (Figure 7). It may take a while for the repository to appear in Software Center.

Removing Software

To remove software, open Software Center again, search for the name of the software, and click *Remove*. The configuration data is thereby unaffected. The advantage is that you don't need to reconfigure the software if you decide you need it again later. If these remnants (such as you might find in the */etc* or */var* directories) were to disappear, use the command line to remove the package.

Figure 7: In this PPA, you'll find the current packages for Raring Ringtail to install the Inkscape design application.

Quick Sources

The Software Center is a graphical interface for apt, the Advanced Packaging Tool, which manages the list of software in the repositories in the background. As already noted, you can also run apt over the command line; many experienced Ubuntu users prefer this method because the commands work with any Ubuntu derivative. Here are four important commands:

```
$ sudo apt-get update
$ sudo apt-cache search (--names-only) inkscape
$ sudo apt-get install inkscape
$ sudo apt-get remove (--purge) inkscape
```

The first command brings the package list up to date, an important thing considering how quickly program versions change.

The next line searches for a package whose name or description includes the term *inkscape*. Add the *--names-only* option (without the parentheses) to search for the package name only if too many hits are returned.

The third line installs the package, and the fourth line removes it. Use the optional *--purge* option to purge the configuration files in the home directory along with the package down to the local configuration that you have to remove yourself.

To remove all the packages installed through a PPA, either use Software Center and choose the PPA next to *Installed*, or stay on the command line, install the *ppa-purge* package, and use the following command (for our Inkscape example):

```
$ sudo ppa-purge ppa:inkscape/stable
```

Single Consignments

Finally, some websites provide single Debian packages to download. To install the software, double-click the DEB package. Ubuntu then calls Software Center and recommends installing the package.

Once you click *OK*, the package appears together with a screenshot and description in Software Center. Finally, download the package with *Install*. •

Info

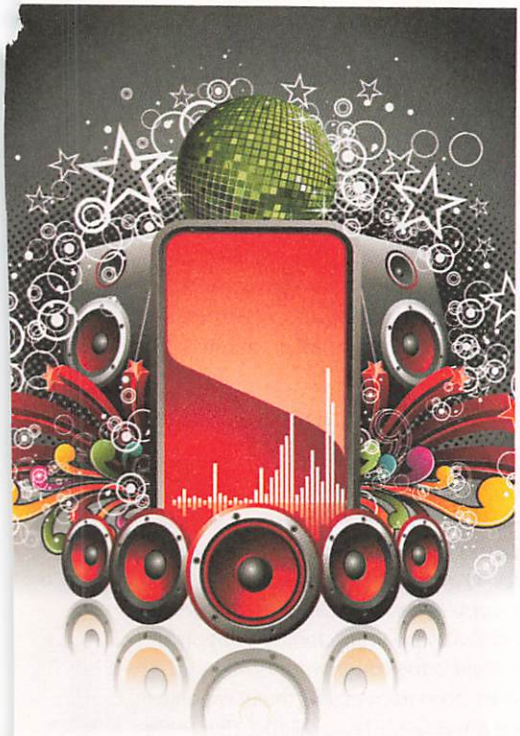
[1] What are the meta, super, and hyper keys? *http://askubuntu.com/questions/19558/what-are-the-meta-super-and-hyper-keys*

[2] Ubuntu's Launchpad platform: *https://launchpad.net*

[3] Opera repository: *http://deb.opera.com/*

[4] VirtualBox project: *http://www.virtualbox.org/wiki/Linux_Downloads*

[5] PPAs in Launchpad: *https://launchpad.net/ubuntu/+ppas*



Multimedia components and proprietary programs

THAT'S ENTERTAINMENT

BY TIM SCHÜRMANN AND MARCEL HILZINGER

Movies, music, and Internet telephony are easy to use in Ubuntu. You just have to load the right software packages onto the computer. We present some favorite multimedia apps.

Playing movies and music, ripping DVDs, and making Skype calls – that's all possible with Ubuntu. Although the Ubuntu installer already provides basic "third-party" multimedia codecs, these options might not fulfill everyone's wishes. At least, the distribution loads the legitimate MP3 codec from Fluendo onto the computer through the *ubuntu-restricted-addons* package, along with components for the GStreamer framework (plugins-ugly, plugins-bad, ffmpeg) and a Flash plugin installer from Adobe.

As a result, you can play MP3 files, burn audio CDs, and view more than 90 movie formats right after installation. However, the installed package is not compliant with the *ubuntu-restricted-extras* package that you should install after the installation. It provides additional codecs for GStreamer, some typical Microsoft fonts (for which you need to sign an EULA first) and the UnRAR unpacker. You'll also need to install the *LAME* library so that you can create and play MP3 files from audio CDs.

GSTREAMER

Generally, numerous programs in Ubuntu play music and movies, but there are also some standard apps. Rhythmbox (Figure 1) is the preinstalled solution for playing music. You can use the software to convert CDs to MP3 format, although RipperX [1] is much better suited for it. Totem is the very simple default video player, and it uses the GStreamer format.

The drop-down menu used to access the videos and podcasts from YouTube and the BBC in the past is missing in the

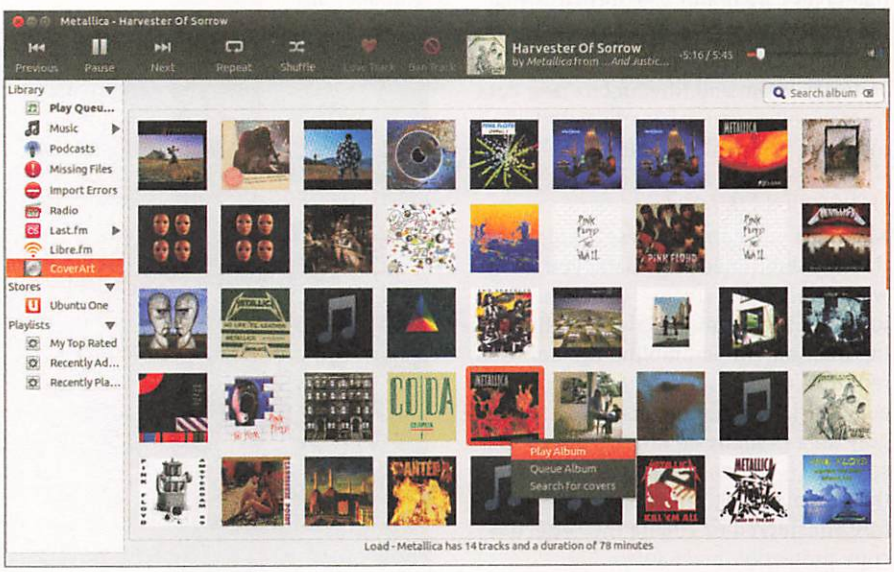


Figure 1: Rhythmbox plays MP3s as well as radio stations, and it provides integrated music shops. Better software exists for converting from audio CDs.

Figure 2: The Gnome developers removed some functions from the current Totem. On the right, the drop-down menu that previously allowed access to YouTube is missing.

current version (Figure 2). Many users, however, still also install VLC: The multimedia player plays everything, provides more features than Totem, and can be extended easily through modules. You can find these modules in Software Center if you search for *vlc-plugin-*.

Brasero is still used as a CD burner, and it provides an integrated cover editor. Burning Blu-rays works only with the commercial Nero Linux 4 product that has a demo version [2].

Running in the background for the previously mentioned programs is GStreamer, which is a modular multimedia framework that's available to many audio and video apps. Thus, many programs don't need to support special codecs because GStreamer provides the playback capability.

To play an MP3 file, Rhythmbox hands over the task to GStreamer, which rummages through its resources for a plugin that takes care of the MP3 format. Using the contained codecs, GStreamer converts the files into audible material (Figure 3). To teach Totem and Rhythmbox new data formats, often you can just install the matching GStreamer plugin.

Dangerous Liaisons

Although you can use these "limited extras" for multimedia formats, you can't watch encrypted DVDs – which covers just about all commercial DVDs. Bypassing the Content Scrambling System (CSS) requires *libdvdcss2*, provided by the VideoLan [3], but this is considered a legal gray area in many countries. Use of this software is banned, because it cancels the CSS playback protection and is therefore considered to violate the "effective copy protection mechanism."

In some countries, you can resort to two commercial Linux programs for legal playback: LinDVD from Corel [4] and the Fluendo DVD player [5], whose developers also participate in GStreamer. You can find the latter in the *For Purchase* repository.

Until a couple of months ago, you could use the Medibuntu repository. It used to house – apart from MPlayer and Mencoder – other codec packages that you needed to play and edit movies in all formats. Medibuntu has since been closed (see the "Bye-Bye, Medibuntu" box), and all its packages have been moved to the main repositories (except for *libdvdcss* mentioned above).

Partners and Shopping Sources

Some proprietary applications like Skype and VMware View Client aren't offered in Ubuntu's standard repositories. Although these programs might be free, they're often under proprietary licenses and are not open source, and, in the case of Skype, very insecure.

These third-party apps can often be found in the *Canonical Partners* repository that you must first activate. In Software Center, select *Edit | Software Sources*, click the *Other Software* tab, then activate the *Canonical Partners* entry. Again, it might take a while before

Bye-Bye, Medibuntu

Medibuntu was closed a few months ago because of lack of maintenance and because it was becoming a security risk for users. So, now, if a movie won't run, usually activating the *Proprietary drivers for devices (restricted)* and *Software restricted by copyright or legal issue (multiverse)* will help. Apart from *libdvdcss2*, these repositories provide MPlayer, Mencoder, fonts for Adobe's Acroread PDF reader, and additional multimedia codecs that are in *non-free-codecs*.

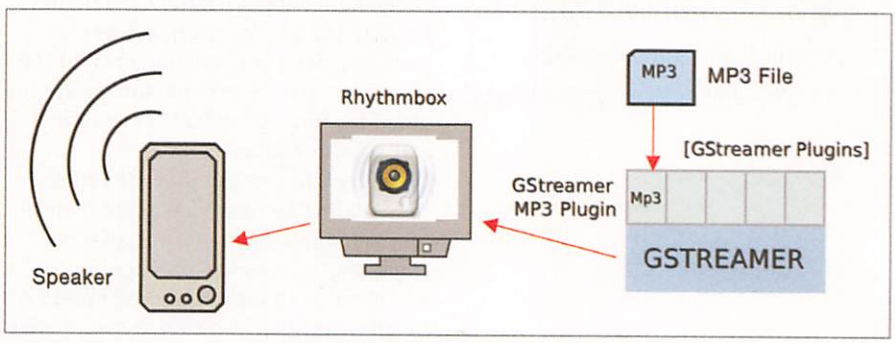


Figure 3: Rhythmbox and Totem rely on the GStreamer framework that provides the necessary plugins with new codecs.

Figure 4: Running under Ubuntu, you'll find not only 1990s-style graphics but also visually appealing games, such as Family Farm.

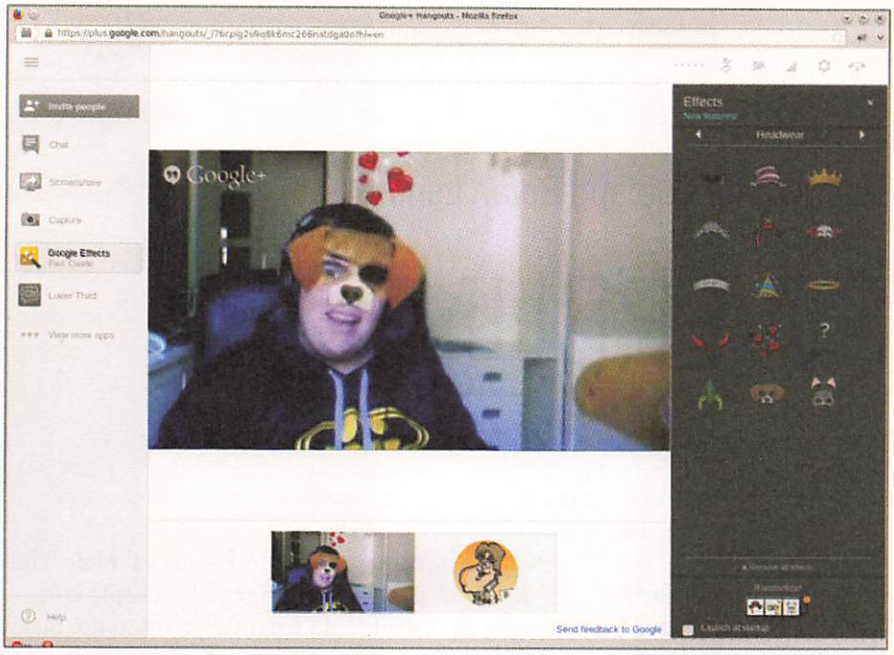


Figure 5: Like Skype, Google Hangout provides video chat capability.

the sources appear in Software Center. The *For Purchase* repository contains mainly software that you can purchase through Software Center – including this magazine in various language editions. Developers usually release the content of these sources on their release date.

When you select the software and click *Purchase*, Software Center links to the Launchpad login page. You can log in to (or register at) the Ubuntu One service and pay via credit card. Apart from official DVD players, you'll also find nifty games such as *Family Farm* (Figure 4), *Oil Rush*, and *World of Goo*.

The same payment plan applies to the repository named *Independent*. There, you'll find mainly apps of external developers that put some minimal time into Ubuntu. Some of them are quite useful, however, because they're meant to handle smaller tasks.

Adobe Products

Loading the *flashplugin-installer* package on the computer brings in the latest Flash versions from the Adobe website. You might have installed Adobe's PDF Reader from the partner repository in the past. But, you will only need it if Ubuntu's Evince built-in PDF viewer breaks down – which happens in rare cases with PDF files with 3D content or complicated graphics with many levels. For security reasons, you should avoid using Adobe Reader.

Skype and Google Video

Even the new version 4 of Skype is in the *Canonical Partners* repository. Apart from videotelephony, it also provides desktop sharing: To make collaboration easier, your conversation partner can take a peek at your desktop.

Thanks to the new version, the image quality of Skype for Linux is gradually matching that of the Windows client. Be aware, however, that Skype is one of the pieces of software that most spies on your communications. If you value your privacy, don't use it.

As an alternative to Skype, you can check out Google's Hangout video platform, which also allows calls among multiple participants (Figure 5). You will need a Google account, so register at the Google website and then go to the left at the top of the Google+ profile.

In the right-hand column, select *Start Hangout* and install a video plugin for your system. You can simply download the corresponding DEB package and double-click the file manager.

Google Earth

If you enjoy using Google Earth [6], you can install it in Ubuntu 14.04, too. Simply download the corresponding Debian package for your architecture, and then go to Software Center and install the *lsb-core* package.

Double-clicking the Google Earth package loads it onto the computer. Note that this step can take a while, mainly because the package grabs additional files from an online server.

Next, make sure you have enough free space reserved. Then, you can start Google Earth by entering *goog* in the Dash. Be forewarned, however: Google Earth is no fun without proper 3D acceleration.

The New Java

To use Java applications, you need a Java Runtime Environment (JRE). Ubuntu does not provide Oracle's proprietary Java version anymore – only the free OpenJDK.

Because some Java programs had problems in the past with OpenJDK, many users often resorted to the proprietary Java (*sun-java*). However, you won't be able to find it anymore because Oracle put it under a nonfree license. Because the proprietary Java is compatible with OpenJDK 7 – the official Java reference – most Java programs should be able to work just fine with the free Java.

You can install all the software from the *openjdk-7-jdk* and *icedtea-7-plugin* packages, and the older version from *openjdk-6-jdk* and *icedtea-6-plugin* using the following command:

```
$ sudo update-alternatives --config java
```

This step activates one of the parallel installed Java versions. If an app refuses to run, it's best to submit a bug report at the Ubuntu wiki [7].

Seek and You Shall Find

You can install many proprietary apps and multimedia codecs from the existing repositories. Numerous proprietary Windows programs also run in the Wine Windows-compatible environment [8]. But always take a look in the Ubuntu package manager, because almost every application has a free alternative. •

Info

[1] RipperX: *http://sourceforge.net/projects/ripperx/*

[2] Nero for Linux: *http://www.nero.com/deu/downloads-linux4-update.php*

[3] libdvdcss at VideoLan: *http://www.videolan.org/developers/libdvdcss.html*

[4] LinDVD: *http://en.wikipedia.org/wiki/LinDVD*

[5] Commercial GStreamer plugins: *http://www.fluendo.com*

[6] Google Earth: *http://www.google.com/earth/index.html*

[7] Reporting bugs: *http://www.ubuntu.com/community/report-problem*

[8] Wine: *http://www.winehq.org/*

Q&A with Stuart Langridge and Bryan Lunduke

ANSWERBUNTU

Stuart Langridge and Bryan Lunduke, of Bad Voltage [1] fame, team up (more or less) to answer all your questions and give you advice regarding installing, configuring, setting up hardware, getting apps, and convincing your grandma to switch to Linux.

MY OWN OS

Hi Brian and Stuart,

I am 14 and have been using Ubuntu for quite some time. But I am bit tired of the constraints. I have tried modding the look and feel and also installed derivatives (Kubuntu, Mint, and so on), but I quickly got bored of all that, too.

What I really want to do is build my own Linux system from scratch. Any advice on how to go about this and where I should start.

My ultimate aim is to build a whole new operating system like Linus did. I am teaching myself C, but do you have any advice on how to do that?

Cheers,

Martin

BRYAN: Building your own OS, from scratch, is awesome. Everyone should try to do it at least once in their life. You probably won't get very far, but what you'll learn in the process will be worth it. Plus... it's just good, nerdy fun.

A great place to start is not actually with Linux at all, but with Minix. Minix is what actually inspired Linus Torvalds to start writing Linux, and it was designed and built by Professor Andrew Tannenbaum to be a fully-functional Unix-like OS, which he could teach people how to build in his operating system classes.

The best guide for doing this is Tannenbaum's book, *Operating Systems: Design and Implementation*, and you will want to get the third edition. This book walks you through creating an OS and all the things you need to know, using Minix as its example, and it's exactly designed for people who want to do what you want to do: make their own OS.

After you're up to speed with Minix, you might then want to look at Linux From Scratch [2], which describes how to build your own Linux system by putting together all the various pieces and parts. It's not quite "build a whole OS from the ground up", but it does show you how and why a modern Linux system is put together, and it also comes with a book (which is free to download and read).

Linux famously differs from Minix in a few ways (in particular, Minix is microkernel-based, which is a term you'll learn about as you explore this stuff, and Tannenbaum and Torvalds had a much-quoted disagreement, back in the early days of Linux, about how Linux is not based around a microkernel).

Understanding those different choices and why they happened will help you make similar decisions about MartinOS as you start to build it and learn about how to build it.

Or, if you want to stay a (quite) bit higher-level, you could simply look at putting together your own Linux distro based on many of the available toolsets, such as RemasterSys or SUSE Studio.

Either way, good luck!

STUART: Although I'm all for not limiting anyone's creativity here, I am inclined to say this: Building a whole OS from scratch is a

gargantuan task and, frankly, most of it is pretty boring, especially because you'll be doing it on your own.

Bryan mentions Minix, and you'll note that Minix is pretty good at basic OS services such as "having a filesystem" and "being able to talk TCP/IP," but if you want anything advanced, you just won't have it.

Talking MTP to your phone? Managing system services so they restart automatically? Having a working X or Mir or Wayland display system on an ATI card? Reading NTFS disks? Handling multiple cores on your Intel core i5 chip?

The reason people use existing things is so you're not spending your time solving problems that are already solved. There's so much cool stuff that the open source ecosystem doesn't have and could do: Why spend your time doing things that have already been done? Yes, you'll learn a lot about how to build an OS, but without attracting lots of others to your project, you're essentially doomed to either build something with ungreat filesystem support and ungreat graphics, or to still be doing this when you're 70 years old, 56 years from now.

You'll make a much more effective contribution, I'd suggest, by picking something else that you'd really like to see on Ubuntu and then making it happen. Build a great video editor, or become an important player in one of the ones that already exists. Make a way to make great games for the Ubuntu phone. Work on the open source nouveau driver and help people with NVidia cards get truly great displays. Make Intel WiDi wireless displays work on an Ubuntu laptop. Build a great Fitbit app.

There's so much that we need and don't yet have, and you have a much better chance of attracting both users and other contributors and really affecting people's lives with a targeted projected than if you write an OS. We already have OSes.

I don't want to rain on your parade, and Bryan's advice is (remarkably) actually pretty good if you really want to build your own operating system, but I'd suggest that you have a better chance of making your mark if you work on something that people will use every day and love you for it.

Office Space

? Hi,

I am considering changing the laptop I use for college over to Ubuntu rather than having partitions, and I have a question about the open office programs. Are they compatible with Microsoft office file types? And, can I start work on an open office file then continue using Word on my desktop? I have heard there are conversion problems sometimes. Is there anything I can do to avoid them?

Rob

A STUART: Rob. Seriously. Worrying about this stuff is so 2005.

What you're really talking about is what is called "Roundtripping," which is being able to work on a document using LibreOffice, make some changes on that document in Microsoft Office, then open it back up in LibreOffice again. Ideally, without anything exploding. The good news is twofold:

First: LibreOffice itself has phenomenally good support for a multitude of office document formats – including multiple revisions of Microsoft Word and Excel. Sure, there was a time, in days long past, when a bit of formatting would get lost or a table messed up, but those days are gone. They've been relegated to whatever discard pile hosts other such software curiosities of the distant past, such as Internet Explorer and Pets.com.

Second: Microsoft Office, itself, has the ability to work with Open Document Format files (the format used by LibreOffice) – been there since Office 2007.

So, you see my son, you have nothing to fear here. Create your text documents. Put in pictures and footnotes. They will still be there on whatever platform you find yourself next.

A BRYAN: My dear friend, Stuart, is a crazy man. Do not listen to anything he says. Ever. I'm going to put this as nicely, and as gently, as possible... You are totally screwed.

The fact of the matter is, no matter how much progress has been made in the realm of document format compatibility between office suites – and, for sure, the progress as been plentiful – the problems are still many-fold. And, they will bite you in the butt every chance they get. Like badgers. Angry little, document-destroying badgers.

For one thing, the fonts are different on different systems. That screws everything up right there. Sure, you can standardize on a small set of freely available OpenType Fonts – and just make sure you have them installed on every system you use – but that will only help so much.

Use a picture with the body text of your document wrapping around it? Have an auto-generated table of contents? Your document is writing checks the office suite can't cash. Sometimes working with an Open Document file in MS Word will work great. Sometimes it really, really won't. Don't worry. It gets worse.

Even if you say "Bag it! I'm just going to standardize on Open Document Format for all of my files and stick to open source office suites!" you are still in for a world of hurt. Save a file in LibreOffice and open it up in Calligra or AbiWord and – assuming the document is more complex than a simple grocery list – some formatting will get, for lack of a better word, really and truly dorked up.

But, all is not lost. Here's what I recommend. Get a thumbdrive (or some cloud storage). Standardize on one office suite (LibreOffice is a dandy choice). Put portable versions of those apps, for Windows, on that thumbdrive – along with all of the fonts you tend to use. If you've got space on that thumbdrive, I also recommend installing a copy of Linux with the same version of your office suite.

Then, wherever you go, shazam. You can edit and work on documents – presentations, your thesis, complex spreadsheets, and what have you – without needing to worry about something getting lost when "roundtripping." •

Info

[1] The Bad Voltage podcast can be heard at: *http://www.badvoltage.org/*

[2] Linux From Scratch: *http://www.linuxfromscratch.org/*

How do you approach an open hardware project? We look at Aaron Seigo's projects Vivaldi and Improv as examples and discover how hard it really is to manage a truly free hardware project.

Developing free hardware using Vivaldi and Improv as examples

Vivaldi Meets Don Quixote

BY FERDINAND THOMMES

Our story starts about two years ago when KDE Plasma developer Aaron Seigo [1] was looking for a way to earn a living. He had been employed by the Trolltech company for KDE development. Then, Trolltech went to Nokia, which in turn sold the Qt framework to Digia.

Shortly thereafter, Seigo (Figure 1) announced a new tablet design based on his and his colleague's specifications. The hardware, based on Zenithink C71 [2], was to be produced in Southeast Asia, with the developers planning heavily on components for which there were free drivers.

The tablet should run Linux in the form of the Mer [3] distribution, a derivative of Meeego, which, in turn, derived from Maemo and Moblin. The user interface should provide KDE Plasma Active [4], a joint project of KDE, OpenSLX [5] and the basysKom [6] company. Plasma Active aimed for a user-friendly interface for mobile platforms with low demands on hardware.

KDE had early on adopted the convergence train of thought but not, like Microsoft or Canonical, by subordinating everything else. As early as 2005, the plans for a common code base across devices began in earnest. Plasma Active was released for the first time in 2011 as a further development of the Plasma desktop and is currently at version 4. Rather than having apps in the foreground, the Activities – well known from KDE SC – stood front and center (Figure 2).

The hardware on what became the Spark 7" tablet was supposed to be tailored just enough for Plasma Active, but not exceedingly so. The first plan included an AMLogic ARM processor with 1 GHz clock rate and a Mali 400 GPU for graphic output. The main memory was 512MB (later extended to 1GB), with 4GB internal memory. The price was to be a maximum 200 euros.

MAKE*PLAY*LIVE

Bad Omen

Had Aaron Seigo known at that time what awaited him, he might have hesitated at that point and saved himself the stress that not only was to affect his health but also his finances. But, if he had, there would be no story to tell.

The first omen was not long in coming. The name Spark, which was supposed to symbolize an inspiration for a free tablet, was similar enough to an existing brand name that the owners contested its use. Exactly who that was remains unknown even today; however, speculation is that it was

Figure 1: KDE Plasma developer and Vivaldi protagonist Aaron Seigo.

Figure 2: Plasma Active is based on KDE Activities.

Oracle and its Sparc. The new name for the mobile device then became Vivaldi.

Developers are still working on the tablet today (Figure 3), which should have hit the market back in 2012, and a release date is still to be determined. Seigo and his colleagues meanwhile introduced an intermediary step by marketing the motherboard designed for Vivaldi as a modular development board with the name Improv [7]. For developing, planning, financing, and marketing of their products, they established the "Make Play Live Partner Network" (MPL [8]) initiative in 2012.

GPL in China

Negotiations with a Chinese manufacturer ran according to the team's plans, and they promised open source drivers for the components. This changed almost overnight when, during a board revision, the manufacturer unexpectedly swapped components. The new parts didn't conform to the requirements, and the manufacturer now could not provide open source drivers.

Hardware manufacturers in Southeast Asia often don't really care that production results in a breached GPL. Additionally, a small project, working with smaller companies, often can't afford to have a per unit production rate below a few thousand, even at the lower limit of their production rate. That greatly reduces the influence a small project can possibly exert.

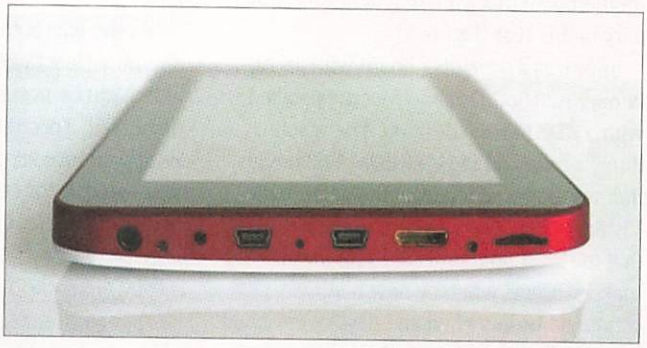


Figure 3: The Vivaldi tablet based on the Zenithink C71A.

Seigo then had to look for a new manufacturer, already well on the way to his deadline. That was not easy to do – even medium-sized manufacturers preferred to produce copies of an already successful product.

Modular Design

After further failed negotiations, the project team decided that they would design the motherboard themselves to ensure that the product met their specifications. This decision meant plenty of extra work but also a significant increase in freedom. It meant that the board components would be defined by them and that the component layout of the mainboard would be freely available.

One outcome that should not be underestimated is the mainboard's modular design. In the Improv, memory and CPU sit piggybacked under the board and can thus be easily replaced. Meanwhile, Seigo, as he describes in his blog [9], also learned to ask the right questions to the manufacturers. The new mainboard design was created together with the Rhombus Tech [10] company and is manufactured by QiMod Technology [11].

After the first mainboard prototype appeared in the spring of 2013, Seigo declared that Vivaldi would be a vastly better tablet than the Spark of the previous year. The PCB design is built around the EOMA-68 [12] (Embedded Open Modular Architecture) CPU card. It measures about 8x5 centimeters, is reminiscent of the 68-pin terminal block of a PCMCIA card (Figure 4), and corresponds to its form factor in many other ways.

Meanwhile, sitting on the EOMA-68 board as a CPU is a 1.2 GHz-clocked Allwinner A10 System on Chip (SoC). Also, the piggyback board houses the RAM, the 8G integrated storage unit, Ethernet chips and the SD card. The sole closed-source component of the CPU is the OpenGL stack – until the Lima driver [14] becomes suitable for production use.

Open Hardware

Open source hardware can be found not only in information technology but also in many other areas. It defines itself by virtue of its license-free blueprints and openly licensed drivers. Various projects strive for different degrees of freedom, depending on feasibility. Among the best known examples of free hardware are Arduino [20], Parallella [21], the US$ 100 laptop OLPC [22], the Open Prosthetics Project [23], and Frankencamera [24].

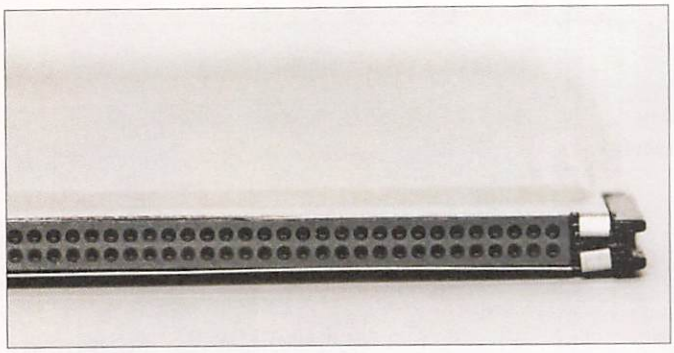


Figure 4: The EOMA68-68 is very similar in shape and connections to a PCMCIA card. (Photo: MPL)

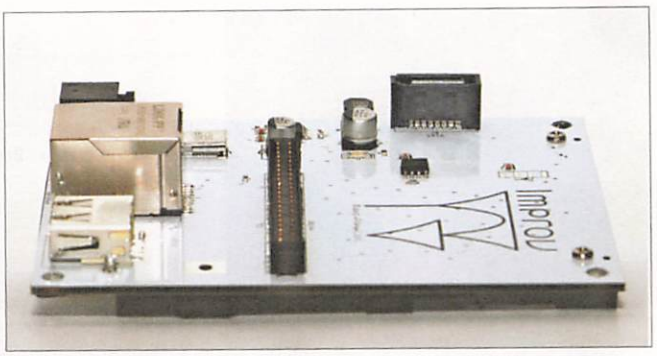


Figure 5: Improv board with DIL connectors. (Photo: MPL)

A Step Forward

Early last summer, the team prepared a YouTube video showing the EOMA-68 card running Debian "Wheezy" [15]. At the end of November, Seigo presented the Improv motherboard that MPL was to deliver in January of this year (Figure 5).

Improv is a developer board with an Allwinner A20 SoC [16] and Mer preinstalled with a KDE Plasma desktop. The whole thing is based on the very EOMA-68 design that was supposed to form the heart of the Vivaldi tablet. The idea behind Improv is that the kit should provide a tool for the makers' scene, whereby new hardware projects could be launched and developers could be helped with professional planning, implementation, and marketing.

Improv includes, apart from the actual CPU board with a 1-GHz ARM dual-core processor, 1GB RAM, 4GB internal memory and a MicroSD card reader. The CPU board sits on a motherboard that provides I/O interfaces for USB, HDMI, SATA, and VGA as well as a dual inline package (DIL) plug connector with 44 pin connectors.

Unfortunately, to date, the boards have not been delivered to first adopters; the orders were considerably fewer than anticipated, whereupon the private investments of the MPL team ran out. The KDE community is running a fundraiser [17] to recoup the costs of producing the board. A KDE webpage [18] also informs the community about the need for open hardware standards and asks for support. Donations above the needed US$ 125,000 will be invested in Improv boards for use in education.

Outlook

Aaron Seigo is an open hardware pioneer whose endurance and stamina can't be appreciated too highly – this goes for his colleagues as well. Starting a free hardware project turns out to be harder than for free software (see also the "Open Hardware" box). The open source community has the means of production in place for software, but free hardware depends on the reliability and credibility of manufacturers in the Far East.

Improv was completed against all odds and now "just" needs to be produced. Vivaldi might be postponed but certainly not dismissed. According to Seigo, it will make its appearance soon. Also, MPL already has two requests for customized Improv boards. Whether the project's rather unspectacular fundraising effort achieves its goal remains to be seen. Something along the lines of Kickstarter or Indiegogo with the motto "Be part of the good fight and tell your friends" might have been more effective.

We wish Aaron and his colleagues well and will report in detail once Improv and Vivaldi become available. You can learn more about Seigo's odyssey with free hardware in his blog. You can find videos for Improv and Vivaldi on the YouTube Make-Play-Live channel [19]. •

Info

[1] Aaron Seigo: *http://en.wikipedia.org/wiki/Aaron_Seigo*

[2] Zenith C71A: *http://www.zenithink.com/Eproducts_C71A.php*

[3] Mer: *http://en.wikipedia.org/wiki/Mer_(software_distribution)*

[4] Plasma Active: *http://plasma-active.org*

[5] OpenSLX: *http://openslx.org/#openslx*

[6] basysKom: *http://www.basyskom.com*

[7] Improv: *http://aseigo.blogspot.de/2013/11/introducing-improv.html*

[8] MPL: *http://makeplaylive.com/index.html#/home*

[9] Aaron's blog: *http://aseigo.blogspot.de/2012/09/a-small-update-on-vivaldi.html*

[10] Rhombus Tech: *http://rhombus-tech.net*

[11] QiMod Tech: *http://www.qimod.com*

[12] EOMA68: *http://elinux.org/Embedded_Open_Modular_Architecture/EOMA-68*

[13] Allwinner A10: *http://en.wikipedia.org/wiki/Allwinner_Technology*

[14] Lima driver: *http://limadriver.org*

[15] EOMA board in video: *https://www.youtube.com/watch?v=Ki7g22ham3s*

[16] Allwinner A20: *http://www.allwinnertech.com/en/clq/processora/2014/0223/265.html*

[17] Supporting project: *http://www.vaultechnology.com/store/donate*

[18] KDE appeal: *http://dot.kde.org/2014/01/22/open-hardware-kde*

[19] MPL YouTube channel: *http://www.youtube.com/channel/UCmCTziEfva7bvBqMrEjzafw*

[20] Arduino: *http://en.wikipedia.org/wiki/Arduino*

[21] Parallella: *http://www.parallella.org/*

[22] OLPC: *http://en.wikipedia.org/wiki/OLPC_XO-1*

[23] Open Prosthetics: *http://openprosthetics.org*

[24] Frankencamera: *http://graphics.stanford.edu/projects/camera-2.0/*

<!DOCTYPE html>
<html>
<body>
<h1> Heroes </h1>
 ...
</body>
</html>

AUTHORS

Jono Bacon	6
Erik Bärwaldt	71
Paul Brown	3, 9
Karsten Günther	28
Marcel Hilzinger	89
Frank Hofmann	42
Kristian Kissling	80, 84, 86
Peter Kreussel	74
Stephan Lamprecht	62
Christoph Langner	89
Stuart Langridge	92
Bryan Lunduke	92
Marcus Nasarek	10
Hartmut Noack	66
Dmitri Popov	58
Karl Sarnow	31, 54
Tim Schürmann	14, 28, 89
Martin Steigerwald	42
Ferdinand Thommes	48, 94
Harald Zisler	20

CONTACT INFORMATION

EDITOR IN CHIEF
Paul Brown, pbrown@ubuntu-user.es

ASSOCIATE EDITOR
Amber Ankerholz

CONTRIBUTING EDITORS
Kristian Kißling

EDITORIAL DIRECTOR
Joe Casad

LOCALIZATION
Bert Gassmann

LAYOUT
Claudio CG

COVER ILLUSTRATION & DESIGN
Fran Sánchez

ADVERTISING
www.ubuntu-user.com/Advertise

NORTH AMERICA
Ann Jesse, ajesse@linuxnewmedia.com
Phone: +1 785 841 8834
Eric Henry, ehenry@linuxnewmedia.com
Phone: +1 785 917 0990

EUROPE
Penny Wilby, pwilby@linuxnewmedia.com
Phone: +44 1787 211100

ALL OTHER COUNTRIES
Petra Jaser, pjaser@medialinx-group.com
Phone: +49 89 9934 1124
Michael Seiter, mseiter@medialinx-group.com
Phone: +49 89 9934 1123

CORPORATE MANAGEMENT (VORSTAND)
Hermann Plank, hplank@medialinx-group.com
Brian Osborn, bosborn@medialinx-group.com

MANAGEMENT NORTH AMERICA
Brian Osborn, bosborn@medialinx-group.com

MARKETING COMMUNICATIONS
Darrah Buren, dburen@linuxnewmedia.com

CUSTOMER SERVICE / SUBSCRIPTION
For USA and Canada:
Email: cs@ubuntu-user.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)
Fax: 1-785-856-3084

For all other countries:
Email: subs@ubuntu-user.com
Phone: +49 89 9934 1167
Fax: +49 89 9934 1199

Ubuntu User • c/o Linux New Media USA
616 Kentucky St. • Lawrence, KS 66044

WWW.UBUNTU-USER.COM – WORLDWIDE

WWW.UBUNTU-USER.COM.AU – AUSTRALIA

WWW.UBUNTU-USER.CO.UK – UNITED KINGDOM

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the DVD provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks ©2014 Linux New Media Ltd.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, emails, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media unless otherwise stated in writing.

Printed in Germany

Distributed by COMAG Specialist, Tavistock Road, West Drayton, Middlesex, UB7 7QE, United Kingdom

Ubuntu User (ISSN 2040-8080) is published quarterly by Linux New Media USA, LLC, 616 Kentucky St, Lawrence, KS 66044, USA, and Linux New Media Ltd, Manchester, England. Company registered in England. Periodicals Postage pending at Lawrence, KS. POSTMASTER: Please send address changes to Ubuntu User, 616 Kentucky St., Lawrence, KS 66044, USA.Linux is a trademark of Linus Torvalds.

Ubuntu® is a trademark of Canonical Ltd and is used under licence from Canonical Ltd. Points of view or opinions in this publication do not necessarily represent the policies or positions of Canonical Ltd or imply affiliation with Ubuntu or www.ubuntu.com. Ubuntu User is not an official Canonical title.

The DVD included in this magazine may contain an unofficial Ubuntu release that is not supported by Canonical Ltd.

Watch for us ...

Want more? Ubuntu User will be back in three months time with more tasty Trusty Tahr treats and tricks. See you then!

Approximate On Sale Date

UK / Europe	May 18
USA / Canada	Jun 13
Australia	Jul 13

www.ubuntu-user.com

vijaiyn / 123RF

4th Annual

THE REGION'S BIGGEST FESTIVAL OF CREATIVITY

Maker Faire® Kansas City

The Greatest Show & Tell on Earth

JUNE 28-29 AT UNION STATION

Featuring Over 350 Maker Booths

INVENTORS • ARTISTS • GARDENERS • FOODISTS • HACKERS • YOUNG MAKERS

ROBOT BUILDERS • SOFTWARE DESIGNERS • DIY-ERS • CRAFTERS

EXCITING!

INNOVATIVE!

EXPLOSIVE!

Maker Faire is a summer family-friendly festival of invention, creativity and resourcefulness, and a celebration of the **Maker Movement.**

Union Station Kansas City hosts the **4th largest Maker Faire** in the United States - THE SHOW-ME STATE'S **MOST INSPIRING EVENT!**

FOR EVENT DETAILS, VISIT:

makerfairekc.com

#makerfairekc

UNION STATION Maker Faire Kansas City MAKE SOMETHING JUNE 28-29

PRESENTING SPONSOR Ewing Marion KAUFFMAN Foundation

IT'S HERE!

A brand new magazine for the Raspberry Pi Community

New Magazine for Raspberry Pi Users!
PACKED with terrific tips, techniques, and projects
Together we'll conquer the world...
raspberry pi
GEEK
WIND POWER
Power your Pi off-grid with an inexpensive wind turbine
COOL PROJECTS:
Build a Rasp Pi laptop
Messaging with Telegram
Arduino Yún: Temperature sensing
Put a LAMP stack on your Pi
Servo Spycam
Roll your own home
Time Ma

Look for us at your local newsstand!

Or find us online at www.raspberry-pi-geek.com